Discrete Applied Mathematics 158 (2010) 1136–1147

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

New results on optimizing rooted triplets consistency☆

Jaroslaw Byrka[a,b,1], Sylvain Guillemot[c], Jesper Jansson[d,*]

[a] Centrum Wiskunde & Informatica (CWI), Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands

[b] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[c] Institut Gaspard Monge-Université Paris-Est, 5 boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée, France

[d] Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form 15 February 2010
Accepted 12 March 2010
Available online 18 April 2010

Keywords:
Phylogenetic tree
Rooted triplet
Supertree
Approximation algorithm
Pseudorandomness
Hardness of approximation

ABSTRACT

A set of phylogenetic trees with overlapping leaf sets is *consistent* if it can be merged without conflicts into a supertree. In this paper, we study the polynomial-time approximability of two related optimization problems called *the maximum rooted triplets consistency problem* (MaxRTC) and *the minimum rooted triplets inconsistency problem* (MinRTI) in which the input is a set $\mathcal{R}$ of rooted triplets, and where the objectives are to find a largest cardinality subset of $\mathcal{R}$ which is consistent and a smallest cardinality subset of $\mathcal{R}$ whose removal from $\mathcal{R}$ results in a consistent set, respectively. We first show that a simple modification to Wu's `Best-Pair-Merge-First` heuristic Wu (2004) [38] results in a bottom-up-based 3-approximation algorithm for MaxRTC. We then demonstrate how any approximation algorithm for MinRTI could be used to approximate MaxRTC, and thus obtain the first polynomial-time approximation algorithm for MaxRTC with approximation ratio less than 3. Next, we prove that for a set of rooted triplets generated under a uniform random model, the maximum fraction of triplets which can be consistent with any phylogenetic tree is approximately one third. We then provide a deterministic construction of a triplet set having a similar property which is subsequently used to prove that both MaxRTC and MinRTI are NP-hard even if restricted to minimally dense instances. Finally, we prove that unless P = NP, MinRTI cannot be approximated within a ratio of $c \cdot \ln n$ for some constant $c > 0$ in polynomial time, where n denotes the cardinality of the leaf label set of $\mathcal{R}$.

1. Introduction

A *supertree method* is a method for merging an input collection of phylogenetic trees on overlapping sets of taxa into a single phylogenetic tree called a *supertree*. An input collection of trees might contain contradictory branching structure, e.g., due to errors in experimental data or because the data originates from different genes, so ideally, a supertree method should merge the input trees while keeping as much of the branching information as possible. Supertree methods are helpful for two main reasons:

- Supertrees can be used to deduce hypothetical evolutionary relationships between taxa which do not occur together in any one of the input trees. For example, in "the tree of life" project, the goal is to build a tree that represents the

☆ An extended abstract of this article was presented in *Proceedings of the 19th Annual International Symposium on Algorithms and Computation* (ISAAC 2008), volume 5369 of *Lecture Notes in Computer Science*, pp. 484–495, Springer-Verlag, 2008.

* Corresponding author.
E-mail addresses: Jaroslaw.Byrka@epfl.ch (J. Byrka), Sylvain.Guillemot@univ-mlv.fr (S. Guillemot), Jesper.Jansson@ocha.ac.jp (J. Jansson).

1 Present address: Institute of Mathematics, EPFL, Station 8 - Bâtiment MA, CH-1015 Lausanne, Switzerland.

0166-218X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2010.03.004

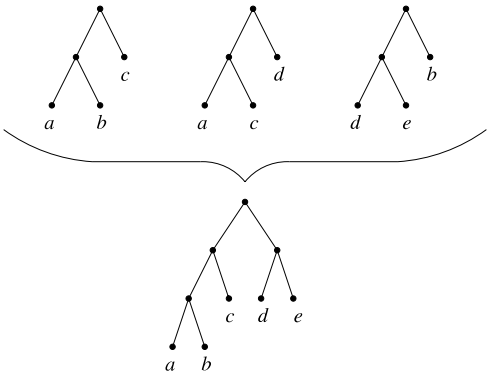

Fig. 1. The set of rooted triplets $\{ab|c, ac|d, de|b\}$ is consistent.

evolutionary history of more than one and a half million species [32], requiring data from an enormous amount of different sources to be combined.

- In the context of inferring a phylogenetic tree from sequence or character data, supertree methods may be convenient in cases where the taxa set is too large for computationally expensive phylogenetic tree reconstruction methods such as maximum likelihood or maximum parsimony to be applied directly. Instead, one may follow a divide-and-conquer approach and first apply an expensive method to infer a collection of highly accurate trees for small, overlapping subsets of the taxa, and then use a computationally cheaper supertree method to merge them [8,17,27].

In this paper, we investigate the computational complexity of some combinatorial problems at the core of rooted supertree methods which involve *rooted triplets*.

1.1. Problem definitions and notation

A *phylogenetic tree* is a rooted, unordered, distinctly leaf-labeled tree in which every internal node has at least two children, and a *rooted triplet* is a binary phylogenetic tree with exactly three leaves. From here on, each leaf in a phylogenetic tree is identified with its label. The unique rooted triplet on leaf set $\{x, y, z\}$ where the lowest common ancestor (lca) of x and y is a proper descendant of the lca of x and z (or equivalently, where the lca of x and y is a proper descendant of the lca of y and z) is denoted by $xy|z$.

For any rooted triplet $xy|z$ and phylogenetic tree T which includes three leaves labeled by x, y, z, if the lca in T of x and y is a proper descendant of the lca of x and z, then $xy|z$ and T are said to be *consistent* with each other; otherwise, $xy|z$ and T are *inconsistent*. A set $\mathcal{R}$ of rooted triplets is *consistent* if there exists a phylogenetic tree T such that every $xy|z \in \mathcal{R}$ is consistent with T. The set of all rooted triplets consistent with a tree T is denoted by $rt(T)$. As a small example, the set of rooted triplets $\{ab|c, ac|d, de|b\}$ is consistent (see Fig. 1), but if $ce|b$ is added to the set then it is easy to show by contradiction that there exists no phylogenetic tree consistent with all four rooted triplets in the resulting set $\{ab|c, ac|d, de|b, ce|b\}$.

Now, we define the three problems RTC, MaxRTC, and MinRTI. For any phylogenetic tree T over a leaf set L and a set $\mathcal{R}$ of rooted triplets over L, let $C(\mathcal{R}, T) = |\mathcal{R} \cap rt(T)|$ and $I(\mathcal{R}, T) = |\mathcal{R} \setminus rt(T)|$, i.e., the number of rooted triplets in $\mathcal{R}$ which are consistent and inconsistent with T, respectively.

- *The rooted triplets consistency problem* (RTC): Given a set $\mathcal{R}$ of rooted triplets with leaf set L, output a phylogenetic tree leaf-labeled by L which is consistent with *every* rooted triplet in $\mathcal{R}$, if one exists; otherwise, output *null*.
- *The maximum rooted triplets consistency problem* (MaxRTC): Given a set $\mathcal{R}$ of rooted triplets with leaf set L, output a phylogenetic tree T leaf-labeled by L which maximizes $C(\mathcal{R}, T)$.
- *The minimum rooted triplets inconsistency problem* (MinRTI): Given a set $\mathcal{R}$ of rooted triplets with leaf set L, output a phylogenetic tree T leaf-labeled by L which minimizes $I(\mathcal{R}, T)$.

The optima for MaxRTC and MinRTI on an instance $\mathcal{R}$ are denoted by $C(\mathcal{R})$ and $I(\mathcal{R})$, respectively. Without loss of generality, we restrict MaxRTC and MinRTI to output binary phylogenetic trees only. We say that an algorithm $\mathcal{A}$ for MaxRTC is an *α-approximation algorithm* (and that the *approximation ratio of* $\mathcal{A}$ is at most α) if, for every input $\mathcal{R}$, the tree output by $\mathcal{A}$ is consistent with at least $\frac{C(\mathcal{R})}{\alpha}$ of the rooted triplets in $\mathcal{R}$. Analogously, an algorithm $\mathcal{B}$ for MinRTI is a *β-approximation algorithm* (and the *approximation ratio of* $\mathcal{B}$ is at most β) if, for every input $\mathcal{R}$, the tree output by $\mathcal{B}$ is inconsistent with at most $I(\mathcal{R}) \cdot \beta$ of the rooted triplets in $\mathcal{R}$. An exact algorithm for either of MaxRTC or MinRTI automatically yields an exact algorithm for the other, but approximation ratios are not preserved, as will be demonstrated in Section 7.

Throughout the paper, we write $n = |L|$ and $k = |\mathcal{R}|$ in the problem definitions above. (Thus, $k = O(n^3)$.) A set $\mathcal{R}$ of rooted triplets over a leaf label set L is called *dense* if it contains at least one rooted triplet labeled by L' for every subset L' of L of cardinality three, and *simple* if it contains at most one rooted triplet for each such subset. $\mathcal{R}$ is *minimally dense* if it is both dense and simple.

For convenience, we also use the following notation. Consider a set L and a total order $>$ on L. For any non-negative integer q, let $[L]^q$ be the set of tuples $(x_1, \ldots, x_q) \in L^q$ with $x_1 > \cdots > x_q$, and let $\langle L\rangle^q$ be the set of tuples $(x_1, \ldots, x_q) \in L^q$ having pairwise distinct coordinates. We will alternatively view a simple triplet set $\mathcal{R}$ on L as a partial function $\mathcal{R} : \langle L\rangle^3 \to \mathbb{Z}_3$ such that for each distinct $x_0, x_1, x_2 \in L$, it holds that $\mathcal{R}(x_0, x_1, x_2) = i$ if and only if $x_{i+1}x_{i+2}|x_i \in \mathcal{R}$, where index addition is modulo 3. Note that $\mathcal{R}$ is fully specified by its restriction to $[L]^3$.

1.2. Motivation

It suffices to consider RTC if no input rooted triplets are in conflict. Furthermore, there exists an efficient algorithm for RTC; see Section 2 below. (Also note that to merge a set of arbitrary, but non-conflicting, phylogenetic trees, one can encode the trees by a small set of rooted triplets and run the algorithm for RTC on this set [17].) However, for applications to phylogenetics, where the underlying data seldom fits nicely into a supertree, it is far more useful to consider optimization versions of RTC. Two natural optimization criteria are: (1) Find a maximum cardinality subset $\mathcal{R}'$ of $\mathcal{R}$ such that $\mathcal{R}'$ is consistent with some phylogenetic tree; and (2) Find a maximum cardinality subset L' of L such that the restriction of $\mathcal{R}$ to L' is consistent with some phylogenetic tree. Criterion (1) is precisely MaxRTC defined above, while criterion (2) leads to *the maximum agreement supertree problem* (MASP), studied in [5,14,22].

The analog of RTC for *unrooted* trees where all of the input trees are *quartets* (unrooted, distinctly leaf-labeled trees each having four leaves and no nodes of degree two) has received a lot of attention; see [27] for references. Interestingly, although RTC is solvable in polynomial time, the quartet consistency problem is NP-hard [33], which indicates that rooted supertree methods may be preferable to unrooted supertree methods from the viewpoint of computational complexity. See also the discussion in [34].

Further motivation for rooted triplet-based supertree methods comes from the fact that reliable rooted triplets can be inferred through maximum likelihood-based methods such as [8] or Sibley–Ahlquist-style DNA–DNA hybridization experiments (see [26]). Moreover, the experimental results in [32] suggest that under certain conditions, rooted triplet-based methods outperform character-based methods even though the latter typically require much more running time.

Rooted triplet-based supertree methods have been applied to real data in [32] (marsupial species as well as rbcL gene data) and in [36] (*Cryptococcus gattii* yeast data).

1.3. New results and organization of the paper

We first give a survey of existing related results in Section 2 and point out that an existing approximation algorithm for MaxRTC named `Min-Cut-Split` is in fact an $(n-2)$-approximation algorithm for MinRTI. Then, in Section 3, we prove that a simple modification to Wu's `Best-Pair-Merge-First` heuristic [38] turns it into an approximation algorithm for MaxRTC with approximation ratio at most 3. In Section 4, we show how any approximation algorithm for MinRTI could be employed to approximate MaxRTC, and use this result to obtain the first polynomial-time approximation algorithm for MaxRTC with approximation ratio smaller than 3. In Section 5, we show that for a set of minimally dense rooted triplets generated under a uniform random model, the maximum fraction of triplets which can be consistent with any phylogenetic tree is approximately $\frac{1}{3}$, and then provide a deterministic construction of a minimally dense triplet set having a similar property. This deterministic construction is later used in Section 6 to prove that MaxRTC and MinRTI are NP-hard even if restricted to minimally dense instances, which is a strengthening of the recent hardness result in [37]. Next, Section 7 proves that (unrestricted) MinRTI cannot be approximated within a ratio of $c \cdot \ln n$ for some constant $c > 0$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$. Finally, Section 8 discusses open problems.

Below, "phylogenetic trees" are referred to as "trees" for short.

2. Previous results

This section lists previously published results concerning the computational complexity of RTC and MaxRTC. To our knowledge, MinRTI has not been studied before.

RTC: Aho et al. [1] introduced RTC and gave a recursive top-down $O(kn)$-time algorithm for the problem. Their algorithm uses a so-called *auxiliary graph*, whose edges are defined by $\mathcal{R}$, to partition the current leaves into *blocks* in such a way that each block consists of all leaves which are in one subtree of the current root, and then recurses on each block.[2] Henzinger et al. [17] reduced the algorithm's complexity to $\min\{O(n + kn^{1/2}),\ O(k + n^2 \log n)\}$ time and $O(n + k \log^3 n)$ expected time by employing dynamic data structures for keeping track of the connected components in the auxiliary graph under batches of edge deletions. By replacing the dynamic graph connectivity data structures with newer ones, such as the data

[2] For any $L' \subseteq L$, the auxiliary graph $\mathcal{G}(\mathcal{R}, L')$ is the undirected graph $\mathcal{G}(\mathcal{R}, L') = (L', E)$, where E contains edge $\{x, y\}$ if and only if there is some $xy|z$ in $\mathcal{R}$ with $x, y, z \in L'$. Then, each connected component of $\mathcal{G}(\mathcal{R}, L')$ induces a block of L'. During execution, if any auxiliary graph having more than one vertex consists of just one connected component then the algorithm returns *null* and terminates.

structure by Holm et al. [18], the running time of the algorithm of Aho et al. can immediately be further improved to $\min\{O(n + k\log^2 n),\ O(k + n^2 \log n)\}$ [22].

Hardness of MaxRTC: MaxRTC was proved to be NP-hard independently in [6,19,38]. Byrka et al. [7] recently observed that the reductions in [6,38] are in fact L-reductions from an APX-hard problem, and therefore that the general (non-dense) case of MaxRTC is APX-hard. van Iersel et al. [37] modified the reductions of [6,38] to prove that MaxRTC remains NP-hard even if restricted to dense input sets.

Exact algorithm for MaxRTC: Wu [38] gave an exact, dynamic-programming algorithm for MaxRTC. It runs in $O((k + n^2)3^n)$ time and $O(2^n)$ space.

Approximation algorithms for MaxRTC: The first polynomial-time approximation algorithms for MaxRTC, henceforth referred to as `One-Leaf-Split` and `Min-Cut-Split`, were presented by Gąsieniec et al. in [13]. Both algorithms are greedy, top-down algorithms.

- `One-Leaf-Split` achieves a constant ratio approximation of MaxRTC; more precisely, it runs in $O((k + n)\log n)$ time and constructs a caterpillar tree which is guaranteed to be consistent with at least one third of the input rooted triplets.
- `Min-Cut-Split` proceeds exactly as the algorithm of Aho et al. [1] with two modifications: (1) the auxiliary graphs are edge-weighed; and (2) if an auxiliary graph has more than one vertex but only one connected component then instead of giving up, `Min-Cut-Split` will find a minimum weight edge cut in the auxiliary graph, delete those edges, and continue.[3] Since deleting an edge from an auxiliary graph corresponds to deleting one or more rooted triplets from $\mathcal{R}$ and since there are at most $n - 2$ recursion levels containing non-trivial auxiliary graphs in the algorithm of Aho et al., it follows that if W denotes the total weight of the input rooted triplets and t the minimum total weight of triplets to remove to achieve consistency then `Min-Cut-Split` constructs a tree which is consistent with a subset of $\mathcal{R}$ whose total weight is $\geq W - (n - 2)t$. This also implies that `Min-Cut-Split` yields an $(n - 2)$-approximation algorithm for MinRTI. `Min-Cut-Split` can be implemented to run in $\min\{O(kn^2 + n^3 \log n),\ O(n^4)\}$ time (see Section 2.4.2 in [20]).

Snir and Rao [32] presented a greedy, top-down, polynomial-time heuristic for MaxRTC called `MXC` which resembles `Min-Cut-Split`. The difference is that `MXC` augments the auxiliary graphs with extra edges, and whenever the algorithm of Aho et al. is stuck with a single connected component, instead of taking a minimum weight edge cut, `MXC` tries to find a cut that *maximizes* the ratio between the extra edges and the ordinary edges. Although the worst-case approximation ratio of `MXC` is unknown, it appears to perform very well on real data [32].[4]

Wu [38] proposed a greedy, bottom-up heuristic for MaxRTC named `Best-Pair-Merge-First`. It runs in polynomial time and is structurally similar to the well-known UPGMA/WPGMA and Neighbor-Joining methods; see, e.g., [10]. No theoretical analysis of the worst-case performance of `Best-Pair-Merge-First` was provided in [38], but Wu demonstrated by extensive simulations that this heuristic performs well in practice (source code in *C* is available from the author's webpage). In another paper, Wu [39] described a heuristic called `Dynamic-Programming-With-Pruning` based on his exact algorithm from [38] which yields a trade-off between the running time and quality of the solution.

A PTAS (polynomial-time approximation scheme) for MaxRTC restricted to dense input sets, based on the work of Jiang et al. [25] for the analogous unrooted (and more difficult) problem, was outlined in [21].

Miscellaneous related results: Several other problems related to RTC and MaxRTC have been studied in the literature. Ng and Wormald [28] showed how to efficiently construct *all* solutions to RTC for any input set of rooted triplets. Gąsieniec et al. [12] considered RTC and MaxRTC for *ordered* trees. He et al. [16] gave algorithms for a variant of RTC/MaxRTC called *the forbidden rooted triplets consistency problem* in which the input consists of a "good" set and a "bad" set of rooted triplets, and the objective is to construct a tree which is consistent with all of the rooted triplets in the good set and none of the rooted triplets in the bad set. Extensions of RTC/MaxRTC to *phylogenetic networks* (generalizations of phylogenetic trees in which certain nodes are allowed to have more than a single parent) have been studied in [7,23,24,35–37]. Finally, an extension of RTC to *multi-labeled phylogenetic trees*, a variant of phylogenetic trees where each leaf label may occur more than once, was recently introduced in [15].

3. A bottom-up 3-approximation algorithm for MaxRTC

Here, we modify Wu's `Best-Pair-Merge-First` heuristic from [38] so that it achieves an approximation ratio of at most 3. Although MaxRTC already admits a polynomial-time 3-approximation algorithm by `One-Leaf-Split`

[3] Semple and Steel [31] later independently developed a heuristic for merging a set of phylogenetic trees with overlapping leaf sets that uses a very similar idea, and Page [29] further modified the heuristic of Semple and Steel.

[4] Incidentally, we would like to point out an apparent error in Lemma 1 in [32]. The lemma claims that for any set of rooted triplets $\mathcal{R}$ (consistent or not), if the auxiliary graph $\mathcal{G}(\mathcal{R}, L)$ is not connected then any optimal tree T^* has a subtree for the leaves at each connected component of $\mathcal{G}(\mathcal{R}, L)$. Now consider $\mathcal{R} = \{ab|c,\ ad|e,\ cf|e\}$ and $T^* = ((((a, b), (c, f)), d), e)$; (in Newick notation). The tree T^* is optimal since it is consistent with all three triplets of $\mathcal{R}$. However, T^* does not have a subtree for the component $\{a, b, d\}$ of $\mathcal{G}(\mathcal{R}, L)$. We note that the argument in their proof can be used to prove the existence of an optimal tree T^* with the stated property, but the property does not hold for any optimal tree.

Algorithm `Modified-BPMF`
Input: A set $\mathcal{R}$ of rooted triplets on a leaf set $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$.
Output: A tree with leaf set L consistent with at least one third of the rooted triplets in $\mathcal{R}$.

1. Construct the set $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, where each S_i is a tree consisting of a leaf labeled by ℓ_i.
2. Repeat $n-1$ times:
   (a) For every $S_i, S_j \in \mathcal{S}$, reset $score(S_i, S_j) := 0$.
   (b) For every $xy|z \in \mathcal{R}$ such that $x \in S_i$, $y \in S_j$, and $z \in S_k$ for three different trees S_i, S_j, S_k, update $score$ as follows:
   $score(S_i, S_j) := score(S_i, S_j) + 2$;
   $score(S_i, S_k) := score(S_i, S_k) - 1$;
   $score(S_j, S_k) := score(S_j, S_k) - 1$.
   (c) Select $S_i, S_j \in \mathcal{S}$ such that $score(S_i, S_j)$ is maximum.
   (d) Create a tree S_k by connecting a new root node to the roots of S_i and S_j.
   (e) $\mathcal{S} := \mathcal{S} \cup \{S_k\} \setminus \{S_i, S_j\}$.
3. Return the tree in $\mathcal{S}$.

Fig. 2. Algorithm `Modified-BPMF`.

(see Section 2), our new result is significant for several reasons. First of all, and perhaps most importantly, `Best-Pair-Merge-First` performs much better than `One-Leaf-Split` in practice [38], but Wu left it as an open problem to derive its approximation ratio. Also, `Best-Pair-Merge-First` uses a bottom-up approach, whereas `One-Leaf-Split` works top-down, and future work may try to incorporate both of these approaches. Finally, `Best-Pair-Merge-First` is faster than `One-Leaf-Split` for large k.

The basic idea of Wu's `Best-Pair-Merge-First` heuristic [38] is to start with n trees, each consisting of a single leaf from L, and repeatedly merge two trees until all leaves are in the same tree. Whenever two trees A and B are to be merged, a new root node is created that represents the merged tree and whose two children are the roots of A and B. A special scoring function determines which pair of trees to merge at each step. `Best-Pair-Merge-First` does the above six times, using six different scoring functions, and returns the best solution among those six.

Our new algorithm is called `Modified-BPMF` and is listed in Fig. 2. Its structure is identical to that of `Best-Pair-Merge-First`, but the approximation ratio of the new algorithm is easier to analyze due to the modified scoring function. Intuitively, in each iteration the algorithm looks for two currently existing trees S_i, S_j whose leaves participate in many rooted triplets of the form $xy|z$ where x belongs to S_i, y belongs to S_j, and z belongs to neither S_i nor S_j. Then, the algorithm joins the roots of S_i and S_j to a new root node. Lemma 1 below proves that at least one third of all input rooted triplets of the form $xy|z$, where exactly one of x, y, z belongs to S_i, exactly one of x, y, z belongs to S_j, and exactly one of x, y, z does not belong to S_i or S_j, will be consistent with the tree just created.

We now analyze the approximation ratio of `Modified-BPMF`. Let T be the final tree returned in Step 3. For any node u of T, let $\mathcal{L}[u]$ be the set of leaf labels in the subtree of T rooted at u. For each internal node u in T, denote the two children of u by u_1 and u_2, and let $\mathcal{R}(u)$ be the subset of $\mathcal{R}$ defined by $\mathcal{R}(u) = \{xy|z \in \mathcal{R} : \exists a, b, c \in \{x, y, z\}$ such that $a \in \mathcal{L}[u_1]$, $b \in \mathcal{L}[u_2]$, and $c \notin \mathcal{L}[u_1] \cup \mathcal{L}[u_2]\}$. Observe that for any two internal nodes u and v, $\mathcal{R}(u)$ and $\mathcal{R}(v)$ are disjoint. Also, each $xy|z \in \mathcal{R}$ belongs to $\mathcal{R}(u)$ for some internal node u. Thus, the internal nodes of T partition $\mathcal{R}$ into disjoint subsets. For each internal node u of T, further partition the set $\mathcal{R}(u)$ into two disjoint subsets $\mathcal{R}(u)'$ and $\mathcal{R}(u)''$ where $\mathcal{R}(u)'$ are the rooted triplets in $\mathcal{R}(u)$ which are consistent with T and $\mathcal{R}(u)'' = \mathcal{R}(u) \setminus \mathcal{R}(u)'$.

Lemma 1. *$|\mathcal{R}(u)'| \geq \frac{1}{3} \cdot |\mathcal{R}(u)|$ for each internal node u of T.*

Proof. Consider the iteration of `Modified-BPMF` ($\mathcal{R}$) in which the node u is created as a new root node for two trees S_i and S_j selected in Steps 2c. Clearly, score$(S_i, S_j) \geq 0$. Moreover, by the definition of score in Steps 2a and 2b and the construction of T, we have score$(S_i, S_j) = 2 \cdot |\mathcal{R}(u)'| - |\mathcal{R}(u)''|$. Since $|\mathcal{R}(u)''| = |\mathcal{R}(u)| - |\mathcal{R}(u)'|$, we obtain $|\mathcal{R}(u)'| \geq \frac{1}{3} \cdot |\mathcal{R}(u)|$. □

Theorem 1. *For any set $\mathcal{R}$ of rooted triplets,* `Modified-BPMF`*($\mathcal{R}$) returns a tree consistent with at least one third of the rooted triplets in $\mathcal{R}$.*

Proof. Follows directly from the fact that $\mathcal{R}$ is partitioned into disjoint subsets by the internal nodes of T, together with Lemma 1. □

`Modified-BPMF` may be implemented as follows. Use $O(k+n^2)$ time for preprocessing to construct an $(n \times n)$-sized table that stores the pairwise scores and an $(n \times n)$-sized table of doubly-linked lists that stores $\mathcal{R}$ (initially, store each $xy|z \in \mathcal{R}$ in the lists for entries (x, y), (x, z), and (y, z)). Then, in each iteration, spend $O(n^2)$ time to find the best pair of trees S_i, S_j to merge, identify $\mathcal{R}(u)$ as the set of all triplets in the list for entry (i, j), and use $O(|\mathcal{R}(u)| + n)$ time to update relevant table entries (remove all $O(|\mathcal{R}(u)|)$ occurrences of triplets belonging to $\mathcal{R}(u)$ from the lists, merge pairs of lists $O(n)$ times, and update $O(|\mathcal{R}(u)|+n)$ scores). Thus, `Modified-BPMF` can be implemented to run in $O((k+n^2)+n\cdot(n^2+n)+\sum_{u\in T}|\mathcal{R}(u)|) = O(k+n^3)$ time. This is faster than the running time of `One-Leaf-Split` for $k = \omega\left(\frac{n^3}{\log n}\right)$.

4. Approximating MaxRTC by using MinRTI

According to the problem definitions, any exact algorithm for MinRTI trivially also provides exact solutions to MaxRTC. In this section, we further investigate this relationship in terms of how approximation algorithms for MinRTI could be used to approximate MaxRTC.

Theorem 2. *Suppose $\mathcal{B}$ is a β-approximation algorithm for* MinRTI *for some $\beta > 1$. Let $\mathcal{A}'$ be the approximation algorithm for* MaxRTC *which returns the best of the two approximate solutions obtained by:* (1) *applying* `Modified-BPMF` *to the input $\mathcal{R}$; and* (2) *applying $\mathcal{B}$ to $\mathcal{R}$ and taking the complement relative to $\mathcal{R}$. Then the approximation ratio of algorithm $\mathcal{A}'$ is at most $\left(3 - \frac{2}{\beta}\right)$.*

Proof. Let $a'(\mathcal{R})$ denote the number of rooted triplets in $\mathcal{R}$ which are consistent with the tree returned by $\mathcal{A}'$. Since $\mathcal{A}'$ returns the best of the two approximate solutions obtained by (1) and (2) above, it always holds that $a'(\mathcal{R}) \geq \frac{1}{3} \cdot k$ (according to Theorem 1) and $a'(\mathcal{R}) \geq k - \beta \cdot (k - C(\mathcal{R}))$. There are two possibilities:

- $k > \frac{3\beta}{3\beta-2} \cdot C(\mathcal{R})$: Then, $a'(\mathcal{R}) \geq \frac{1}{3} \cdot k > \frac{1}{3} \cdot \frac{3\beta}{3\beta-2} \cdot C(\mathcal{R}) = \frac{1}{3-\frac{2}{\beta}} \cdot C(\mathcal{R})$.
- $k \leq \frac{3\beta}{3\beta-2} \cdot C(\mathcal{R})$: In this case, $a'(\mathcal{R}) \geq k - \beta \cdot (k - C(\mathcal{R})) = \beta \cdot C(\mathcal{R}) - (\beta - 1) \cdot k \geq \beta \cdot C(\mathcal{R}) - (\beta - 1) \cdot \frac{3\beta}{3\beta-2} \cdot C(\mathcal{R}) = \left(\beta - \frac{(\beta-1)\cdot 3\beta}{3\beta-2}\right) \cdot C(\mathcal{R}) = \frac{1}{3-\frac{2}{\beta}} \cdot C(\mathcal{R})$.

In both cases, we have $a'(\mathcal{R}) \geq \frac{1}{3-\frac{2}{\beta}} \cdot C(\mathcal{R})$. □

By plugging in `Min-Cut-Split` (see Section 2) into Theorem 2, one directly obtains:

Corollary 1. MaxRTC *admits a polynomial-time $\left(3 - \frac{2}{n-2}\right)$-approximation algorithm.*

5. Random and pseudorandom minimally dense triplet sets

This section examines properties of minimally dense sets of rooted triplets constructed in a random or pseudorandom fashion. We first show that for a triplet set, generated under a uniform random model, the maximum fraction of triplets that can be consistent is approximately one third. We then adapt a construction from [2] to obtain a deterministic construction of a triplet set having a similar property.

5.1. Uniform random model

Let L be a set of n elements. Consider a minimally dense set $\mathcal{R}$ of rooted triplets on L generated by the following random model: for each $t \in [L]^3$, $\mathcal{R}(t)$ is a uniformly chosen random element of $\mathbb{Z}_3$. The next theorem shows that the maximum fraction of triplets from $\mathcal{R}$ which can be consistent is approximately $1/3$.

Theorem 3. *Let $\mu = \frac{1}{3}\binom{n}{3}$. Let $\delta(n)$ be any function such that $\delta(n) = \Omega\left(\frac{\log n}{n}\right)$. With high probability: $C(\mathcal{R}) < (1 + \delta(n))\mu$.*

Proof. Fix δ. Given a binary tree T on L, we compute the probability that $C(\mathcal{R}, T)$ deviates from its expectation by a factor $1 + \delta$. Given a triplet $t \in rt(T)$, denote by $\chi(\mathcal{R}, t)$ the indicator variable which equals 1 if $t \in \mathcal{R}$ and 0 otherwise. Observe that $C(\mathcal{R}, T)$ is a sum of i.i.d. random variables:

$$C(\mathcal{R}, T) = \sum_{t \in rt(T)} \chi(\mathcal{R}, t).$$

Since $\mathbb{E}[C(\mathcal{R}, T)] = \mu$, a straightforward application of Chernoff bounds yields:

$$\mathbb{P}[C(\mathcal{R}, T) > (1 + \delta)\mu] \leq \exp(-c\mu\delta^2)$$

for some constant c. Now, apply union bounds to obtain:

$$\mathbb{P}[C(\mathcal{R}) > (1 + \delta)\mu] \leq \sum_{T} \mathbb{P}[C(\mathcal{R}, T) > (1 + \delta)\mu] \leq 2^{n \log n} \exp(-c\mu\delta^2).$$

Observe that if $\delta = \Omega\left(\frac{\log n}{n}\right)$ then $c\mu\delta^2 = \Omega(n \log^2 n)$; hence the above expression tends to 0 as n tends to infinity. □

$l_0l_2|l_1,\ l_0l_3|l_1,\ l_2l_3|l_0,\ l_1l_2|l_3,\ l_1l_4|l_0,\ l_0l_2|l_4,\ l_1l_2|l_4,$
$l_0l_3|l_4,\ l_1l_3|l_4,\ l_3l_4|l_2,\ l_0l_1|l_5,\ l_0l_2|l_5,\ l_1l_2|l_5,\ l_0l_3|l_5,$
$l_3l_5|l_1,\ l_2l_5|l_3,\ l_4l_5|l_0,\ l_1l_5|l_4,\ l_2l_5|l_4,\ l_4l_5|l_3,\ l_0l_1|l_6,$
$l_0l_2|l_6,\ l_2l_6|l_1,\ l_3l_6|l_0,\ l_1l_6|l_3,\ l_2l_6|l_3,\ l_0l_6|l_4,\ l_1l_6|l_4,$
$l_4l_6|l_2,\ l_3l_4|l_6,\ l_0l_6|l_5,\ l_5l_6|l_1,\ l_2l_5|l_6,\ l_3l_5|l_6,\ l_4l_5|l_6.$

Fig. 3. The 35 triplets of the minimally dense triplet set $\mathcal{R}_7$ over the label set $\mathbb{Z}_7$ identified with $\{l_0, l_1, l_2, l_3, l_4, l_5, l_6\}$.

5.2. Deterministic construction

We now describe a deterministic construction of a minimally dense random-like triplet set. It uses the following algebraic construction which generalizes the construction of Ailon and Alon [2] by introducing an additive parameter q. The original construction of [2] provides an s-coloring of the hyperedges of the complete r-uniform hypergraph, with the pseudorandom properties summarized in Lemma 2 below.

Definition 1. Consider integers $r, s > 1$, a prime p with $s \mid p-1$, and an element $q \in \mathbb{Z}_p$. Let g be a generator of $\mathbb{Z}_p^*$, let H be the subgroup of $\mathbb{Z}_p^*$ generated by g^s, and for each $0 \le i < s$ let H_i be the coset Hg^i.

For an element $j \in \mathbb{Z}_p^*$, define $[j]_p^s = i$ if $j \in H_i$, and define $[0]_p^s = 0$. Define $\phi_{p,q}^{r,s} : \mathbb{Z}_p^r \to \{0, \ldots, s-1\}$ so that for each $j = (j_1, \ldots, j_r) \in \mathbb{Z}_p^r$, $\phi_{p,q}^{r,s}(j) = [j_1 + \cdots + j_r + q]_p^s$.

Furthermore, for any set $A \subset \mathbb{Z}_p^r$, and $0 \le j < s$, write:

$$n_j(A) = |\{i \in A : \phi_{p,q}^{r,s}(i) = j\}|.$$

The following lemma from [2] states that if A arises from a cartesian product and if $|A|$ is large enough, then the fraction of hyperedges of A which have color j is approximately $1/s$.

Lemma 2 (*[2], Lemma 2.5*). *Let $A_1, \ldots, A_r$ be subsets of $\mathbb{Z}_p$, and let $A = \{i \in [\mathbb{Z}_p]^r : i_j \in A_j, j = 1 \cdots r\}$. Then for all $1 \le j < s$,*

$$|n_j(A) - |A|/s| \le c_r |A|^{1/2} (\log |A|)^{r-1} p^{(r-1)/2}$$

for some global $c_r > 0$ that depends only on r.

While our construction in Definition 1 differs from [2] by using an additional parameter q, a careful inspection of the proof of Lemma 2.3 of [2] shows that it holds if we replace the sum $i_1 + \cdots + i_r$ by the sum $i_1 + \cdots + i_r + q$. It is then easy to see that Lemmas 2.4 and 2.5 of [2] remain correct with the new definition of $[.]_p^s$. Thus, our Lemma 2 is correct in this setting.

We apply the construction of Definition 1 with $r = s = 3$ to obtain a minimally dense triplet set $\mathcal{R}_p$ on $\mathbb{Z}_p$ with random-like properties. More precisely, we define $\mathcal{R}_p$ so that for each $(x, y, z) \in [\mathbb{Z}_p]^3$, $\mathcal{R}_p(x, y, z) = \phi_{p,0}^{3,3}(x, y, z)$. (Recall that $x > y > z$ holds according to the definitions in Section 1.1.)

As an example, Fig. 3 depicts the triplet set $\mathcal{R}_p$ for $p = 7$ which consists of $\binom{7}{3} = 35$ triplets. In the construction, we chose $g = \overline{3}$ as a generator of $\mathbb{Z}_7^*$ so that $g^s = g^3 = \overline{6}$, $H_0 = H = \{\overline{1}, \overline{6}\}$, $H_1 = \{\overline{3}, \overline{4}\}$, and $H_2 = \{\overline{2}, \overline{5}\}$. For instance, the triplet $l_0l_2|l_1$ is obtained as follows: $[\overline{0} + \overline{1} + \overline{2}]_7^3 = 1$, hence $\mathcal{R}_7(l_2, l_1, l_0) = 1$, which implies that $l_0l_2|l_1 \in \mathcal{R}_7$. By running the exact algorithm for MaxRTC, one finds that $C(\mathcal{R}_7) = 21$ and that there are only two optimal solutions: the trees $((((l_0, l_2), l_1), l_6), ((l_4, l_5), l_3))$; and $((((l_0, l_2), l_6), l_1), ((l_4, l_5), l_3))$; (in Newick notation).

The next theorem shows that $\mathcal{R}_p$ is random-like for large values of p: when p is large enough, every binary tree with leaves labeled by $\mathbb{Z}_p$ is consistent with approximately one third of the triplets in $\mathcal{R}_p$. Its proof relies on Lemma 2 and makes use of the new additive parameter q.

Theorem 4. *For any binary tree T on $\mathbb{Z}_p$, it holds that $\left|C(\mathcal{R}_p, T) - \frac{1}{3}\binom{p}{3}\right| \le cp^{5/2} \log p$ for some constant c.*

Proof. Fix $z \in \mathbb{Z}_p$. Let $L_{z,1}, \ldots, L_{z,m}$ be the clusters hanging along the path in T from z to the root; these sets form a partition of $\mathbb{Z}_p \setminus \{z\}$. For each $i \in [m]$, let $n_{z,i}$ be the number of triplets of $\mathcal{R}_p \cap rt(T)$ of the form $xy|z$ with $x, y \in L_{z,i}$. We then have: $C(\mathcal{R}_p, T) = \sum_{z \in \mathbb{Z}_p} \sum_i n_{z,i}$.

Fix $i \in [m]$, and let $A_{z,i} = [L_{z,i}]^2$. We will show that $|n_{z,i} - |A_{z,i}|/3| \le cp^{3/2} \log p$. Define the sets $L_{z,i}^{(1)} = \{x \in L_{z,i} : x < z\}$ and $L_{z,i}^{(2)} = \{x \in L_{z,i} : x > z\}$, and partition $A_{z,i}$ into three sets $A_{z,i}^{(1)} = [L_{z,i}^{(1)}]^2, A_{z,i}^{(2)} = L_{z,i}^{(2)} \times L_{z,i}^{(1)}$, and $A_{z,i}^{(3)} = [L_{z,i}^{(2)}]^2$. Next, define $f : [\mathbb{Z}_p \setminus \{z\}]^2 \to \mathbb{Z}_3$ by setting $f(x, y) = \phi_{p,z}^{2,3}(x, y)$. For any $A \subset [\mathbb{Z}_p]^2$, $j \in \mathbb{Z}_3$, let $n_j'(A) = |\{i \in A : f(i) = j\}|$. We then have:

$$n_{z,i} = n_1'(A_{z,i}^{(1)}) + n_2'(A_{z,i}^{(2)}) + n_3'(A_{z,i}^{(3)}).$$

Since $f = \phi_{p,z}^{2,3}$, Lemma 2 applies and yields the following inequality: for each $j \in \{1, 2, 3\}$,

$$|n_j'(A_{z,i}^{(j)}) - |A_{z,i}^{(j)}|/3| \le c' |A_{z,i}^{(j)}|^{1/2} (\log |A_{z,i}^{(j)}|) p^{1/2}$$

for some constant c'. By using the triangle inequality and by summing over index j, we obtain:

$$|n_{z,i} - |A_{z,i}|/3| \leq c|A_{z,i}|^{1/2}(\log |A_{z,i}|)p^{1/2}$$

for some constant c. Let $S = \sum_{z\in\mathbb{Z}_p}\sum_i |A_{z,i}|$ and $S' = \sum_{z\in\mathbb{Z}_p}\sum_i |A_{z,i}|^{1/2}$. By summing over indices z, i in the previous inequality, we obtain:

$$|C(\mathcal{R}_p, T) - S/3| \leq cS'p^{1/2}\log p.$$

We conclude by observing that $S = \binom{p}{3}$ and that $S' \leq p^2$ (this last inequality following from the fact that $\sum_i |A_{z,i}|^{1/2} \leq \sum_i |L_{z,i}| = p - 1$ holds for any fixed z). □

Corollary 2. $\left|C(\mathcal{R}_p) - \frac{1}{3}\binom{p}{3}\right| \leq cp^{5/2}\log p$.

Note that $|\mathcal{R}_p| = \binom{p}{3}$. Thus, $C(\mathcal{R}_p)$ will be close to $\frac{1}{3}\cdot|\mathcal{R}_p|$ for large p.

6. NP-hardness of MaxRTC and MinRTI for minimally dense inputs

Our first hardness result concerns the computational complexity of MaxRTC and MinRTI for *minimally dense* inputs. It is based on the deterministic construction of a minimally dense random-like triplet set given in Section 5 and is a non-trivial strengthening of the NP-hardness proof for *dense* inputs found in [37].

Theorem 5. *The restriction of* MaxRTC *to minimally dense instances is* NP*-hard.*

Proof. We reduce the general (non-dense) case of MaxRTC (which is already known to be NP-hard [6,19,38]) to the minimally dense case, following an approach inspired by [2,3]. Starting with an arbitrary instance, the approach consists in replicating each label p times (which is called *inflating* the instance), and making the resulting instance dense by adding a pseudorandom triplet set. Formally, the reduction proceeds as follows. Consider a triplet set $\mathcal{R}$ on L given as an instance of MaxRTC. Let $n = |L|$, let p be a prime number, and let $L' = \{x_i : x \in L, i \in \mathbb{Z}_p\}$. Define the minimally dense triplet set $\mathcal{R}'$ on L' by:

1. if $\mathcal{R}(x, y, z)$ is defined then $\mathcal{R}'(x_i, y_j, z_k) = \mathcal{R}(x, y, z)$;
2. if $\mathcal{R}(x, y, z)$ is undefined and i, j, k are distinct then $\mathcal{R}'(x_i, y_j, z_k) = \mathcal{R}_p(i, j, k)$, where $\mathcal{R}_p$ is the minimally dense triplet set defined in Section 5;
3. otherwise, $\mathcal{R}'(x_i, y_j, z_k)$ is an arbitrary element of $\mathbb{Z}_3$.

For $i \in \{1, 2, 3\}$, let $\mathcal{R}'_i$ be the triplet set defined by condition i, so that $\mathcal{R}' = \mathcal{R}'_1 \cup \mathcal{R}'_2 \cup \mathcal{R}'_3$. Observe that $\mathcal{R}'$ is obtained by inflating $\mathcal{R}$, resulting in $\mathcal{R}'_1$, and completing the instance by a pseudorandom triplet set $\mathcal{R}'_2$ and an arbitrary triplet set $\mathcal{R}'_3$. The correctness of the reduction follows from the fact that inflating the instance multiplies the measure by a factor p^3, while completing the triplet set introduces noise which can be made small by proper choice of p, in such a way that an optimum for $\mathcal{R}$ can be recovered from an optimum for $\mathcal{R}'$. To be more precise, let us introduce the following notation. Let $N_1 = n$, let $N_2 = n(n-1)$, let N_3 be the number of triples $\{x, y, z\}$ such that $\mathcal{R}(x, y, z)$ is undefined, and let $N = N_1 + 8N_2 + 27N_3$. It can be shown that the following relations hold:

1. $C(\mathcal{R}'_1) = p^3 C(\mathcal{R})$;
2. for each binary tree T on L', $\left|C(\mathcal{R}'_2, T) - N\binom{p}{3}/3\right| \leq cNp^{5/2}\log p$;
3. for each binary tree T on L', $C(\mathcal{R}'_3, T) \leq Np^2$.

where 2, follows from the pseudorandomness of $\mathcal{R}_p$ stated in Theorem 4.

It follows that $C(\mathcal{R}')$ is an approximation of $C(\mathcal{R}_1) + Np^3/18 = p^3C(\mathcal{R}) + Np^3/18$ within an additive error of $cNp^{5/2}\log p$, for some constant c. Dividing by $p^3/18$, we obtain

$$\left|\frac{18C(\mathcal{R}')}{p^3} - (18C(\mathcal{R}) + N)\right| \leq c'Np^{-1/2}\log p$$

for some constant c'. Since $N = O(n^3)$, we can choose p polynomially bounded in terms of n such that the right-hand side is less than $\frac{1}{2}$, implying that $\left\lfloor \frac{18C(\mathcal{R}')}{p^3} \right\rfloor = 18C(\mathcal{R}) + N$. □

Corollary 3. *The restriction of* MinRTI *to minimally dense instances is* NP*-hard.*

Proof. Follows from Theorem 5 and the fact that MinRTI is the supplementary problem of MaxRTC. □

7. Polynomial-time inapproximability of MinRTI for general inputs

We now establish a hardness of approximation result for MinRTI in the general case, namely a logarithmic inapproximability by reduction from Minimum Hitting Set.

Theorem 6. MinRTI *is not approximable within* $c \cdot \ln n$ *for some constant* $c > 0$ *unless* $\mathsf{P} = \mathsf{NP}$.

The proof of this theorem is carried out in two steps (Lemmas 3 and 4). We first consider a *weighted version* of MinRTI, called MinRTI-W, defined as follows: Given a label set L, let $\mathcal{T}(L)$ be the set of all possible rooted triplets over L. A *weighted triplet set* on L is a function $\mathcal{R} : \mathcal{T}(L) \to \mathbb{N}$, and given a binary tree T on L, we define $I(\mathcal{R}, T) = \sum_{t \in \mathcal{T}(L) \setminus rt(T)} \mathcal{R}(t)$. The MinRTI-W problem takes a weighted triplet set $\mathcal{R}$ on L and seeks a binary tree T on L such that $I(\mathcal{R}, T)$ is minimum.

We describe a measure-preserving reduction from the Minimum Hitting Set problem (defined in, e.g., [11]) to MinRTI-W in Lemma 3, followed by a measure-preserving reduction from MinRTI-W to MinRTI in Lemma 4. Since the Minimum Hitting Set problem is equivalent to the Minimum Set Cover problem (as can be seen by reversing the roles of subsets and elements), and it is known that Minimum Set Cover cannot be approximated within a ratio of $c \cdot \ln n$ for some constant $c > 0$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$ [4,30], the hardness of approximation of MinRTI then follows.

When referring to the hitting set problem, we will use the language of hypergraphs, calling the subsets hyperedges and the elements vertices. For the clarity of the argument, we will assume that each (hyper)edge contains at least 3 vertices. This assumption corresponds to restricting a Minimum Set Cover instance to have each element being a member of at least 3 subsets. Note that such a restricted version of the Minimum Set Cover problem is as difficult to approximate as the standard Minimum Set Cover problem.

Lemma 3. *There exists a measure-preserving reduction from* Minimum Hitting Set *to* MinRTI-W.

Proof. Let $H = (V, E)$ be a hypergraph given as an instance of Minimum Hitting Set. For each $e \in E$, $v \in e$, define a weighted triplet set $\mathcal{R}_{e,v}$ as follows. Observe that we may assume $|e| > 2$, since otherwise dummy vertices may be inserted. The label set $L_{e,v}$ consists of:

(i) a label a,
(ii) a label c_v,
(iii) for each $u \in e$, a label b_u,
(iv) for each $j = 1, \ldots, |e| - 2$, a label $d_{e,v,j}$.

Assuming any fixed ordering of the vertices in V, for any subset $S \subset V$ we let $S(i)$ denote the i-th element of S in the ordering. Also, we will use e^{-v} to denote $e \setminus \{v\}$. We now define the weighted triplets $\mathcal{R}_{e,v}$ on labels $L_{e,v}$. The only triplets with positive weight in $\mathcal{R}_{e,v}$ are:

(i) the triplet $t_v = b_v c_v | a$,
(ii) the triplet $c_v d_{e,v,1} | b_{e^{-v}(1)}$,
(iii) for each $1 < j \leq |e^{-v}| - 1 = |e| - 2$, a triplet $d_{e,v,j-1} d_{e,v,j} | b_{e^{-v}(j)}$,
(iv) the triplet $d_{e,v,m-1} a | b_{e^{-v}(m)}$, where $m = |e^{-v}|$.

The triplet t_v has weight 1, and the other triplets have a large weight $W > n$. Note that the triplet set $\mathcal{R}_{e,v}$ has exactly one triplet of the form $..|b_u$ for each $u \in e \setminus \{v\}$.

For a given $e \in E$, we define the weighted triplet set $\mathcal{R}_e$ on the label set $L_e = \cup_{v \in e} L_{e,v}$ as the common extension of the functions $\mathcal{R}_{e,v}$ ($e \in E$, $v \in e$). We then define the weighted triplet set $\mathcal{R}$ on the label set $L = \cup_{e \in E} L_e$ as the common extension of the functions $\mathcal{R}_e$ ($e \in E$).

The intuition behind this construction is as follows. For a given $e \in E$, $v \in e$, the auxiliary graph $\mathcal{G}(\mathcal{R}_{e,v}, L_{e,v})$ (see the first footnote in Section 2 for a definition of the auxiliary graph $\mathcal{G}$) is a path P_v formed by the vertices b_v, c_v, $d_{e,v,1}, \ldots, d_{e,v,m-1}$, a. Next, $\mathcal{G}(\mathcal{R}_e, L_e)$ is the union of these graphs, hence it consists of a central vertex a with outgoing paths P_v ($v \in e$). This graph is connected, hence $\mathcal{R}_e$ is not consistent. However, removing one of the triplets t_v makes the resulting auxiliary graph disconnected, by disconnecting b_v from the rest of the graph. In fact, the removal of a triplet t_v makes the resulting triplet set consistent, since when recursing on the subset formed by $L_e \setminus \{b_v\}$, an edge is destroyed on each path $P_{v'}$ with $v' \neq v$, and the subsequent recursive calls can also be seen to succeed. This implies that, in order to obtain consistency of the triplet set $\mathcal{R}$, for each edge e at least one of the triplets t_v has to be violated, and the corresponding vertices form a hitting set of H. For an illustration of the idea, see the example in Fig. 4.

We now prove formally that the reduction is measure-preserving. We use the parenthesized Newick-notation without semicolons to represent binary trees, and let $\langle T_1, \ldots, T_n \rangle$ denote the tree obtained by starting with a binary caterpillar tree[5] with leaves labeled $1, \ldots, n$ and substituting leaf i by tree T_i.

Claim 1. Given a binary tree T on L, we can construct in polynomial time a hitting set of H of size $\leq I(\mathcal{R}, T)$.

[5] A *binary caterpillar tree* is a rooted binary tree in which every internal node has at least one child that is a leaf.

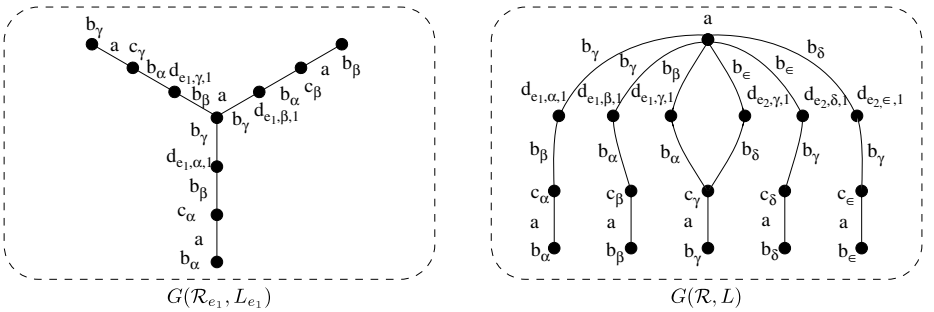

Fig. 4. Example: Two auxiliary graphs corresponding to the gadget constructed for the hypergraph $H = (\{\alpha, \beta, \gamma, \delta, \epsilon\}, \{e_1, e_2\})$, $e_1 = \{\alpha, \beta, \gamma\}$, and $e_2 = \{\gamma, \delta, \epsilon\}$.

Proof of Claim 1. Since triplets other than t_v for some $v \in V$ have large weight W, if one of them is not consistent with T, then trivially $|V| \leq I(\mathcal{R}, T)$ and we output the set V. Suppose only some triplets t_v are inconsistent. For each $e \in E$, since $\mathcal{G}(\mathcal{R}_e, L_e)$ is connected, there must exist $v \in e$ such that $t_v \notin rt(T)$. Let S be the set of elements $v \in V$ such that $t_v \notin rt(T)$. It remains to observe that S is a hitting set of H and $|S| \leq I(\mathcal{R}, T)$. □

Claim 2. Given a hitting set S of H, we can construct in polynomial time a binary tree T on L such that $I(\mathcal{R}, T) = |S|$.

Proof of Claim 2. Given a hitting set S, we partition the set of labels. For a fixed vertex $v \in V$ we divide the labels indexed by v (except b_v for $v \in S$) into σ_v and σ'_v.

- If $v \in S$, set

$$\sigma_v = \emptyset,$$
$$\sigma'_v = \{c_v\} \cup \{d_{e,v,j} : e \ni v, j = 1, \ldots, |e| - 2\}.$$

- If $v \notin S$, then for each edge $e \ni v$ consider the other vertices in the edge e^{-v}. Recall that there is a predefined ordering of the elements in e^{-v}. Since S is a hitting set, $S \cap e^{-v}$ is not empty. Define $j_{v,e}$ to be the index of the smallest element from e^{-v} which is also in S, so that $e^{-v}(j_{v,e}) \in S$ and $e^{-v}(j) \notin S$ for all $j = 1, \ldots, j_{v,e} - 1$. Define

$$\sigma_v = \{b_v, c_v\} \cup \{d_{e,v,j} : e \ni v, 1 \leq j < j_{v,e}\},$$
$$\sigma'_v = \{d_{e,v,j} : e \ni v, j_{v,e} \leq j \leq |e^{-v}|\}.$$

Suppose that $V = \{v_1, \ldots, v_n\}$. Define trees τ, τ', τ'' as follows.

- For each $v \in V$, let τ_v be an arbitrary binary tree on the label set σ_v. Let $\tau = \langle \tau_{v_1}, \ldots, \tau_{v_n} \rangle$.
- For each $v \in V$, let τ'_v be an arbitrary binary tree on the label set σ'_v. Let $\tau' = \langle a, \tau'_{v_1}, \ldots, \tau'_{v_n} \rangle$.
- Let τ'' be an arbitrary binary tree on the label set $\{b_v : v \in S\}$.

Next, define $T = ((\tau, \tau'), \tau'')$. We proceed to show that, for each $e \in E$, $v \in e$, the triplets of $\mathcal{R}_{e,v}$ are consistent with T, except the triplets t_v ($v \in S$).

- Triplets defined by condition (i): if $v \notin S$, b_v, c_v appear in τ, while a appears in τ', hence $b_v c_v | a$ is consistent with T.
- Triplets defined by condition (ii): we show that $c_v d_{e,v,1} | b_{e^{-v}(1)}$ is consistent with T, by considering three subcases:
  - if $v \in S$: then c_v, $d_{e,v,1}$ appear in τ', while $b_{e^{-v}(1)}$ appears in τ or τ'';
  - if $v \notin S$ and $j_{v,e} > 1$: then c_v, $d_{e,v,1}$ appear in τ_v, while $b_{e^{-v}(1)}$ appears in $\tau_{e^{-v}(1)}$ (with obviously $e^{-v}(1) \neq v$);
  - if $v \notin S$ and $j_{v,e} = 1$: then c_v appears in τ, $d_{e,v,1}$ appears in τ' and $b_{e^{-v}(1)}$ appears in τ'' (we have $e^{-v}(1) \in S$ since $j_{v,e} = 1$).
- Triplets defined by condition (iii): we show that $d_{e,v,j-1} d_{e,v,j} | b_{e^{-v}(j)}$ is consistent with T, by considering four subcases:
  - if $v \in S$: then $d_{e,v,j-1}$, $d_{e,v,j}$ appear in τ', while $b_{e^{-v}(j)}$ appears in τ or τ'';
  - if $v \notin S$ and $j < j_{v,e}$: then $d_{e,v,j-1}$, $d_{e,v,j}$ appear in τ_v, while $b_{e^{-v}(j)}$ appears in $\tau_{e^{-v}(j)}$ (with $e^{-v}(j) \neq v$);
  - if $v \notin S$ and $j = j_{v,e}$: then $d_{e,v,j-1}$ appear in τ, $d_{e,v,j}$ appear in τ', and $b_{e^{-v}(j)}$ appears in τ'' (we have $e^{-v}(j) \in S$ since $j = j_{v,e}$);
  - if $v \notin S$ and $j > j_{v,e}$: then $d_{e,v,j-1}$, $d_{e,v,j}$ appear in τ', while $b_{e^{-v}(j)}$ appears in τ or τ''.
- Triplets defined by condition (iv): we show that $d_{e,v,m-1} a | b_{e^{-v}(m)}$ for $m = |e^{-v}|$ is consistent with T, by considering three subcases:
  - if $v \in S$: then $d_{e,v,m-1}$, a appear in τ', while $b_{e^{-v}(m)}$ appears in τ or τ'';
  - if $v \notin S$ and $j_{v,e} < m$: then $d_{e,v,m-1}$, a appear in τ', while $b_{e^{-v}(m)}$ appears in τ or τ'';
  - if $v \notin S$ and $j_{v,e} = m$: then $d_{e,v,m-1}$ appears in τ, a appears in τ', and $b_{e^{-v}(m)}$ appears in τ'' (we have $e^{-v}(m) \in S$ since $j_{v,e} = m$).

We conclude that $I(\mathcal{R}, T) = |S|$ as claimed. □

Lemma 4. *There exists a measure-preserving reduction from* MinRTI-W *to* MinRTI.

Proof. Let $\mathcal{T}(L)$ denote the set of all possible triplets on the label set L. Given a weighted triplet set $\mathcal{R}$ on L, construct an unweighted triplet set $\mathcal{R}'$ on the label set L' where L' is obtained from L by adjoining labels t_i for each $t \in \mathcal{T}(L)$, $1 \leq i \leq \mathcal{R}(t)$, and the triplet set $\mathcal{R}'$ consists of the triplets $xt_i|z, yt_i|z$ for all $t = xy|z \in \mathcal{T}(L)$, $1 \leq i \leq \mathcal{R}(t)$. The next two claims imply that the reduction is measure-preserving.

Claim 1. Given a binary tree T on L, we can construct in polynomial time a binary tree T' on L' such that $I(\mathcal{R}', T') = I(\mathcal{R}, T)$.

Proof of Claim 1. Let $<$ be an arbitrary total order on L. Starting with T, we define T' as follows: for each triplet $t = xy|z \in \mathcal{T}(L)$ with $x < y$, for each $1 \leq i \leq \mathcal{R}(t)$, insert t_i as a sibling of x. We claim that $I(\mathcal{R}', T') = I(\mathcal{R}, T)$. Indeed, consider $t = xy|z \in \mathcal{T}(L)$ with $x < y$, then: (i) if $xy|z \in rt(T)$, then for each $1 \leq i \leq \mathcal{R}(t)$, $xt_i|z, yt_i|z \in rt(T')$, hence the contribution of these triplets to $I(\mathcal{R}', T')$ is 0; (ii) if $xz|y \in rt(T)$, then for each $1 \leq i \leq \mathcal{R}(t)$, $xt_i|z \in rt(T')$ but $yt_i|z \notin rt(T')$, hence the contribution of these triplets to $I(\mathcal{R}', T')$ is equal to $\mathcal{R}(t)$; (iii) if $yz|x \in rt(T)$, the reasoning is similar.

Claim 2. Given a binary tree T' on L', we can construct in polynomial time a binary tree T on L such that $I(\mathcal{R}, T) \leq I(\mathcal{R}', T')$.

Proof of Claim 2. Consider the tree $T = T'|L$ obtained from T' by removing the additional labels t_i. Take any triplet $t = xy|z \in \mathcal{T}(L) \setminus rt(T)$, i.e., a triplet that contributes to $I(\mathcal{R}, T)$. Suppose that there exists i such that $xt_i|z \in rt(T')$ and $yt_i|z \in rt(T')$. Observe that it implies $xy|z \in rt(T')$, which contradicts our choice of t. Hence, no i such that $xt_i|z \in rt(T')$ and $yt_i|z \in rt(T')$ exists. Therefore for each $1 \leq i \leq \mathcal{R}(t)$, one of $xt_i|z, yt_i|z$ is not in $rt(T')$, and the contribution of these triplets to $I(\mathcal{R}', T')$ is at least $\mathcal{R}(t)$. Summing up over $t = xy|z \in \mathcal{T}(L) \setminus rt(T)$ we get $I(\mathcal{R}, T) \leq I(\mathcal{R}', T')$. □

Notice that the measure-preserving reduction from Minimum Hitting Set to MinRTI also yields a hardness result for the parameterized version of MinRTI. The parameterized class W[2] admits the parameterized Hitting Set problem as a complete problem, under parameterized reductions [9]. We obtain:

Corollary 4. *The parameterized version of* MinRTI *is* W[2]*-hard.*

This result shows that the problem is unlikely to be fixed-parameter tractable (i.e., solvable in $f(k)n^c$ time, where k is the parameter and n is the instance size, and where $f : \mathbb{N} \to \mathbb{N}$ is an arbitrary function and $c \in \mathbb{N}$ is a constant).

8. Concluding remarks

The following table summarizes what is currently known about the polynomial-time approximability of MaxRTC and MinRTI:

	Negative results	Positive results
MaxRTC:		
General inputs	APX-hard [7]	$\left(3 - \frac{2}{n-2}\right)$-approx. (Section 4)
Dense inputs	NP-hard [37]	PTAS [21]
Minimally dense inputs	NP-hard (Section 6)	PTAS (↑)
MinRTI:		
General inputs	Inapprox. $c \cdot \ln n$ (Section 7)	$(n-2)$-approx. ([13] & Section 2)
Dense inputs	NP-hard [37]	$(n-2)$-approx. (↑)
Minimally dense inputs	NP-hard (Section 6)	$(n-2)$-approx. (↑)

Significantly, MaxRTC can be approximated within a constant ratio of 3 in polynomial time whereas MinRTI cannot be approximated within a ratio of $c \cdot \ln n$ for some constant $c > 0$ in polynomial time, unless P = NP. (A similar situation occurs for many other pairs of optimization problems such as the maximum independent set problem which is very hard to approximate [40] and its "supplement", the minimum vertex cover problem, which can be trivially approximated within a ratio of 2 [11].)

The main open problem for MaxRTC is to determine whether it admits a constant-ratio polynomial-time approximation algorithm whose approximation ratio is asymptotically better than 3. Since MaxRTC is APX-hard [7], a PTAS is unlikely. Note that both of the 3-approximation algorithms `One-Leaf-Split` from [13] and `Modified-BPMF` in Section 3 always output a solution consistent with at least one third of the input rooted triplets and that in this sense, they are worst-case optimal [13].

We would also like to know: Is it possible to achieve a polynomial-time, polylogarithmic approximation algorithm for MinRTI? Furthermore, is there a polynomial-time, constant-ratio approximation for dense inputs? In particular, how well do the existing approximation algorithms for MaxRTC perform on MinRTI restricted to dense inputs?

Finally, it would be useful to improve the efficiency of Wu's exact, exponential-time algorithm for MaxRTC given in [38]. It seems easy for certain restricted types of trees such as caterpillar trees, but much more challenging for the general case.

Acknowledgements

We thank Leo van Iersel, Judith Keijsper, Steven Kelk, Kazuya Maemura, Hirotaka Ono, Kunihiko Sadakane, and Leen Stougie for helpful comments. The third author was funded by the Special Coordination Funds for Promoting Science and Technology, Japan.

References

[1] A.V. Aho, Y. Sagiv, T.G. Szymanski, J.D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM Journal on Computing 10 (1981) 405–421.
[2] N. Ailon, N. Alon, Hardness of fully dense problems, Information and Computation 205 (2007) 1117–1129.
[3] N. Alon, Ranking tournaments, SIAM Journal on Discrete Mathematics 20 (2006) 137–142.
[4] N. Alon, D. Moshkovitz, S. Safra, Algorithmic construction of sets for k-restrictions, ACM Transactions on Algorithms 2 (2006) 153–177.
[5] V. Berry, F. Nicolas, Maximum agreement and compatible supertrees, Journal of Discrete Algorithms 5 (2007) 564–591.
[6] D. Bryant, Building trees, hunting for trees, and comparing trees: theory and methods in phylogenetic analysis, Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, 1997.
[7] J. Byrka, P. Gawrychowski, K.T. Huber, S. Kelk, Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks, Journal of Discrete Algorithms 8 (2010) 65–75.
[8] B. Chor, M. Hendy, D. Penny, Analytic solutions for three taxon ML trees with variable rates across sites, Discrete Applied Mathematics 155 (2007) 750–758.
[9] R. Downey, M. Fellows, Parameterized Complexity, Springer-Verlag, 1999.
[10] J. Felsenstein, Inferring Phylogenies, Sinauer Associates, Inc., 2004.
[11] M. Garey, D. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, New York, 1979.
[12] L. Gąsieniec, J. Jansson, A. Lingas, A. Östlin, Inferring ordered trees from local constraints, in: Proceedings of Computing: the 4th Australasian Theory Symposium, CATS'98, in: Australian Computer Science Communications, vol. 20(3), Springer-Verlag, Singapore, 1998, pp. 67–76.
[13] L. Gąsieniec, J. Jansson, A. Lingas, A. Östlin, On the complexity of constructing evolutionary trees, Journal of Combinatorial Optimization 3 (1999) 183–197.
[14] S. Guillemot, V. Berry, Fixed-parameter tractability of the maximum agreement supertree problem, in: Proceedings of the 18th Annual Symposium on Combinatorial Pattern Matching, CPM 2007, in: LNCS, vol. 4580, Springer-Verlag, 2007, pp. 274–285.
[15] S. Guillemot, J. Jansson, W.-K. Sung, Computing a smallest multi-labeled phylogenetic tree from rooted triplets, in: Proceedings of the 20th Annual International Symposium on Algorithms and Computation, ISAAC 2009, in: LNCS, vol. 5878, Springer-Verlag, 2009, pp. 1205–1214.
[16] Y.J. He, T.N.D. Huynh, J. Jansson, W.-K. Sung, Inferring phylogenetic relationships avoiding forbidden rooted triplets, Journal of Bioinformatics and Computational Biology 4 (2006) 59–74.
[17] M.R. Henzinger, V. King, T. Warnow, Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology, Algorithmica 24 (1999) 1–13.
[18] J. Holm, K. de Lichtenberg, M. Thorup, Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity, Journal of the ACM 48 (2001) 723–760.
[19] J. Jansson, On the complexity of inferring rooted evolutionary trees, in: Proceedings of the Brazilian Symposium on Graphs, Algorithms, and Combinatorics, GRACO 2001, in: Electronic Notes in Discrete Mathematics, vol. 7, Elsevier, 2001, pp. 121–125.
[20] J. Jansson, Consensus algorithms for trees and strings, Ph.D. Thesis, Lund Institute of Technology, Lund University, Lund, Sweden, 2003.
[21] J. Jansson, A. Lingas, E.-M. Lundell, A triplet approach to approximations of evolutionary trees, Poster H15 presented at RECOMB 2004, 2004.
[22] J. Jansson, J.H.-K. Ng, K. Sadakane, W.-K. Sung, Rooted maximum agreement supertrees, Algorithmica 43 (2005) 293–307.
[23] J. Jansson, N.B. Nguyen, W.-K. Sung, Algorithms for combining rooted triplets into a galled phylogenetic network, SIAM Journal on Computing 35 (2006) 1098–1121.
[24] J. Jansson, W.-K. Sung, Inferring a level-1 phylogenetic network from a dense set of rooted triplets, Theoretical Computer Science 363 (2006) 60–68.
[25] T. Jiang, P. Kearney, M. Li, A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application, SIAM Journal on Computing 30 (2001) 1942–1961.
[26] S. Kannan, E. Lawler, T. Warnow, Determining the evolutionary tree using experiments, Journal of Algorithms 21 (1996) 26–50.
[27] P. Kearney, Phylogenetics and the quartet method, in: T. Jiang, Y. Xu, M.Q. Zhang (Eds.), Current Topics in Computational Molecular Biology, The MIT Press, Massachusetts, 2002, pp. 111–133.
[28] M.P. Ng, N.C. Wormald, Reconstruction of rooted trees from subtrees, Discrete Applied Mathematics 69 (1996) 19–31.
[29] R.D.M. Page, Modified mincut supertrees, in: Proceedings of the 2nd Workshop on Algorithms in Bioinformatics, WABI 2002, in: LNCS, vol. 2452, Springer-Verlag, 2002, pp. 537–552.
[30] R. Raz, S. Safra, A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP, in: Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, STOC'97, ACM, 1997, pp. 475–484.
[31] C. Semple, M. Steel, A supertree method for rooted trees, Discrete Applied Mathematics 105 (2000) 147–158.
[32] S. Snir, S. Rao, Using max cut to enhance rooted trees consistency, IEEE/ACM Transactions on Computational Biology and Bioinformatics 3 (2006) 323–333.
[33] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, Journal of Classification 9 (1992) 91–116.
[34] M. Steel, A.W.M. Dress, S. Böcker, Simple but fundamental limitations on supertree and consensus tree methods, Systematic Biology 49 (2000) 363–368.
[35] T.-H. To, M. Habib, Level-k phylogenetic networks are constructable from a dense triplet set in polynomial time, in: Proceedings of the 20th Annual Symposium on Combinatorial Pattern Matching, CPM 2009, in: LNCS, vol. 5577, Springer-Verlag, 2009, pp. 275–288.
[36] L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, F. Hagen, T. Boekhout, Constructing level-2 phylogenetic networks from triplets, IEEE/ACM Transactions on Computational Biology and Bioinformatics 6 (2009) 667–681.
[37] L. van Iersel, S. Kelk, M. Mnich, Uniqueness, intractability and exact algorithms: reflections on level-k phylogenetic networks, Journal of Bioinformatics and Computational Biology 7 (2009) 597–623.
[38] B.Y. Wu, Constructing the maximum consensus tree from rooted triples, Journal of Combinatorial Optimization 8 (2004) 29–39.
[39] B.Y. Wu, Constructing evolutionary trees from rooted triplets, Journal of Information Science and Engineering 20 (2004) 181–190.
[40] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory of Computing 3 (2007) 103–128.