New Results on Optimizing Rooted Triplets Consistency

Jaroslaw Byrka[1], Sylvain Guillemot[2], and Jesper Jansson[3]

[1] Centrum Wiskunde & Informatica (CWI), Kruislaan 413, NL-1098 SJ Amsterdam, Netherlands and Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands. E-mail: J.Byrka@cwi.nl
[2] INRIA Lille - Nord Europe, 40 avenue Halley, Bât.A, Park Plaza, 59650 Villeneuve d'Ascq, France. E-mail: Sylvain.Guillemot@lifl.fr
[3] Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan. E-mail: Jesper.Jansson@ocha.ac.jp

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 [25] results in a bottom-up-based 3-approximation 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 smaller 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 tree is approximately one third, and 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 MinRTI cannot be approximated within a ratio of $\Omega(\log n)$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$.

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. 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 $xy|z$. If $xy|z$ is an embedded subtree of a tree T, i.e., if the lca of x and y is a proper descendant of the lca of x and z in T, 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 $rt(T)$.

Let L be a set of leaf labels. A set $\mathcal{R}$ of rooted triplets over 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.

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, define $C(\mathcal{R}, T) = |\mathcal{R} \cap rt(T)|$ and $I(\mathcal{R}, T) = |\mathcal{R} \setminus rt(T)|$ (the number of rooted triplets in $\mathcal{R}$ which are consistent and inconsistent with T, respectively). *The rooted triplets consistency problem* (RTC) is: 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) is: 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) is: 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.

An algorithm $\mathcal{A}$ for MaxRTC is an *α-approximation algorithm* (and 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 6.

Denote $n = |L|$ and $k = |\mathcal{R}|$ in the problem definitions above. (Thus, $k = O(n^3)$.) Consider a set L and a total order $>$ on L. For any non-negative integer q, let $[L]^q$ be the set of tuples $(x_1, ..., x_q) \in L^q$ with $x_1 > ... > x_q$, and let $\langle L \rangle^q$ be the set of tuples $(x_1, ..., x_q) \in L^q$ 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 $x_{i+1}x_{i+2}|x_i \in \mathcal{R}$. Note that $\mathcal{R}$ is fully specified by its restriction to $[L]^3$.

1.2 New results and organization of the paper

We first give a survey of existing results in Section 2. Then, in Section 3, we prove that a simple modification to Wu's `Best-Pair-Merge-First` heuristic [25] 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 rooted triplets generated under a uniform random model, the maximum fraction of triplets which can be consistent with any tree is approximately $\frac{1}{3}$, and provide a deterministic construction of a triplet set having a similar property. Section 6 proves that MaxRTC and MinRTI are NP-hard even if restricted to minimally dense instances, which is a strengthening of the result in [14]. Section 6 also proves that (unrestricted) MinRTI cannot be approximated within a ratio of $\Omega(\log n)$ in polynomial time, unless P = NP. Finally, Section 7 discusses open problems.

2 Previous results

This section lists known 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. It 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[1]. Henzinger *et al.* [11] 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 structure by Holm *et al.* [12], the running time of the algorithm of Aho *et al.* can immediately be further improved to $\min\{O(n + k \log^2 n), O(k + n^2 \log n)\}$ [17].

Hardness of MaxRTC: MaxRTC was proved to be NP-hard independently in [4], [15], and [25]. [5] recently observed that the reductions in [4] and [25] are in fact L-reductions from an APX-hard problem, and hence that the general (non-dense) case of MaxRTC is APX-hard. [14] modified the reductions of [4] and [25] to prove that MaxRTC remains NP-hard even if restricted to dense inputs.

Exact algorithm for MaxRTC: Wu [25] gave an exact, dynamic-programming algorithm for MaxRTC. It runs in $O((k + n^2)3^n)$ time and $O(2^n)$ space.

[1] 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.

Approximation algorithms for MaxRTC: The first polynomial-time approximation algorithms for MaxRTC, henceforth referred to as `One-Leaf-Split` and `Min-Cut-Split`, were proposed by Gąsieniec *et al.* in [9]. 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. On the other hand, `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[2]. 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.

Snir and Rao [24] 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 [24].

Wu [25] gave a greedy, bottom-up, polynomial-time heuristic for MaxRTC named `Best-Pair-Merge-First` which is structurally similar to the well-known UPGMA/WPGMA and Neighbor-Joining methods, described in detail in, e.g., [7]. It starts with singleton sets, each containing a single leaf label, and repeatedly merges two sets until all leaf labels are in the same set; whenever two sets A and B are merged, a new internal node is created that represents the merged set and whose two children are the (already existing) nodes representing A and B. A special scoring function determines which pair of sets 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. No theoretical analysis of the worst-case performance of `Best-Pair-Merge-First` was provided in [25], but Wu demonstrated by extensive simulations that this heuristic performs well in practice (source code in C is available from the author's webpage).

A PTAS for MaxRTC restricted to dense inputs, based on the work of [20] for the analogous unrooted (and more difficult) problem, was outlined in [16].

[2] Semple and Steel [23] later independently developed a heuristic for merging a set of phylogenetic trees with overlapping leaf sets that uses a very similar idea, and Page [22] further modified the heuristic of Semple and Steel.

Miscellaneous related results: Other problems related to RTC/MaxRTC have been studied in the literature. Ng and Wormald [21] showed how to efficiently construct *all* solutions to RTC for any input set of rooted triplets. Gąsieniec *et al.* [8] considered RTC and MaxRTC for *ordered* trees. He *et al.* [10] 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. Recently, extensions of RTC/MaxRTC to *phylogenetic networks* (generalizations of phylogenetic trees in which certain nodes are allowed to have more than one parent) have been studied in [5, 13, 14, 18, 19].

3 A bottom-up 3-approximation algorithm for MaxRTC

Here, we modify Wu's `Best-Pair-Merge-First` heuristic [25] to achieve an approximation ratio of at most 3. Although MaxRTC already admits a polynomial-time 3-approximation by `One-Leaf-Split` (see Section 2), our new result is significant because `Best-Pair-Merge-First` outperforms `One-Leaf-Split` in practice [25] and Wu left it as an open problem to derive its approximation ratio. Also, `Best-Pair-Merge-First` uses a bottom-up approach while `One-Leaf-Split` works top-down, and future work may try to incorporate both approaches.

The new algorithm is called `Modified-BPMF` and is listed in Fig. 1. Intuitively, in each iteration it 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, and then merges S_i and S_j.

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. 1. Algorithm `Modified-BPMF`.

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 Step 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 Lemma 1 and the fact that $\mathcal{R}$ is partitioned into disjoint subsets by the internal nodes of T. □

`Modified-BPMF` can be implemented to run in $O(k+n^3)$ time by using $O(k+n^2)$ time for preprocessing and then spending $O(n^2)$ time in each iteration to find the best pair of trees to merge and $O(n^2+|\mathcal{R}(u)|)$ time in each iteration to update all relevant scores. This is faster than `One-Leaf-Split` for $k = \omega(n^3/\log n)$.

4 Approximating MaxRTC by using MinRTI

In this section, we investigate how approximation algorithms for MinRTI can 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 $(3 - \frac{2}{\beta})$.*

Proof. Let $a'(\mathcal{R})$ be the number of rooted triplets in $\mathcal{R}$ 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, $a'(\mathcal{R}) \geq \frac{1}{3} \cdot k$ (according to Theorem 1) and $a'(\mathcal{R}) \geq k - \beta \cdot (k - C(\mathcal{R}))$ always hold. 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, $a'(\mathcal{R}) \geq k - \beta \cdot (k - C(\mathcal{R})) = \beta \cdot C(\mathcal{R}) - (\beta - 1) \cdot k \geq \beta \cdot C(\mathcal{R}) - (\beta - 1) \cdot \frac{3\beta}{3\beta-2} \cdot C(\mathcal{R}) = (\beta - \frac{(\beta-1)\cdot 3\beta}{3\beta-2}) \cdot C(\mathcal{R}) = \frac{1}{3-\frac{2}{\beta}} \cdot C(\mathcal{R})$.

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 obtains:

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

5 Random and pseudorandom triplet sets

This section examines properties of minimally dense sets of 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.

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 following theorem shows that the maximum fraction of triplets 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* $\delta(n) = \Omega(\frac{\log n}{n})$*. 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 $\delta = \Omega(\frac{\log n}{n})$ then $c\mu\delta^2 = \Omega(n \log^2 n)$, hence the above expression tends to 0 as n tends to infinity. □

In the rest of this section, we describe a deterministic construction of a minimally dense random-like triplet set. It uses the following algebraic construction which generalizes the construction of [2] by introducing an additive parameter q. The construction provides an s-coloring of the hyperedges of the complete r-uniform hypergraph, with the pseudorandom properties stated 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* $i \in [s]$ *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* $\phi_{p,q}^{r,s} : \mathbb{Z}_p^r \to [s]$ *so that for each* $j = (j_1, ..., j_r) \in \mathbb{Z}_p^r$*,* $\phi_{p,q}^{r,s}(j) = [j_1 + ... + j_r + q]_p^s$*.*

Furthermore, for any set $A \subset \mathbb{Z}_p^r$, and $j \in [s]$, 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 $j \in [s]$ is approximately $1/s$.

Lemma 2 ([2]). *Let $A_1, ..., A_r$ be subsets of $\mathbb{Z}_p$, and let $A = \{i \in [\mathbb{Z}_p]^r : i_j \in A_j, j = 1...r\}$. Then for all $j \in [s]$, $|n_j(A) - |A|/s| \leq c_r |A|^{1/2} (\log |A|)^{r-1} p^{(r-1)/2}$ for some global $c_r > 0$ that depends only on r.*

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 distinct $x, y, z \in \mathbb{Z}_p$, $\mathcal{R}_p(x, y, z) = \phi_{p,0}^{3,3}(x, y, z)$. The next theorem shows that $\mathcal{R}_p$ is random-like: every binary tree is consistent with approximately one third of the triplets in $\mathcal{R}_p$. The proof relies on Lemma 2.

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

Proof. Fix $z \in \mathbb{Z}_p$. Let $L_{z,1}, ..., L_{z,m}$ be the clusters hanging along the path in T from z to the root; these sets form a partition of $\mathbb{Z}_p \backslash \{z\}$. 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 $|n_{z,i} - |A_{z,i}|/3| \leq cp^{3/2} \log p$. 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 $A_{z,i}^{(1)} = [L_{z,i}^{(1)}]^2$, $A_{z,i}^{(2)} = L_{z,i}^{(2)} \times L_{z,i}^{(1)}$, $A_{z,i}^{(3)} = [L_{z,i}^{(2)}]^2$. Next, define $f : [\mathbb{Z}_p \backslash \{z\}]^2 \to \mathbb{Z}_3$ by setting $f(x, y) = \phi_{p,z}^{2,3}(x, y)$. Given $A \subset [\mathbb{Z}_p]^2, j \in \mathbb{Z}_3$, we set $n'_j(A) = |\{i \in A : f(i) = j\}|$. 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 $j \in \{1, 2, 3\}$, $|n'_j(A_{z,i}^{(j)}) - |A_{z,i}^{(j)}|/3| \leq c' |A_{z,i}^{(j)}|^{1/2} (\log |A_{z,i}^{(j)}|) p^{1/2}$ 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 for fixed z, $\sum_i |A_{z,i}|^{1/2} \leq \sum_i |L_{z,i}| = p - 1$). □

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

6 Hardness results

6.1 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 in [14].

Theorem 5. MaxRTC *restricted 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 [4, 15, 25]) 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 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 $\mathcal{R}'$. More precisely, 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:

1. $C(\mathcal{R}'_1) = p^3 C(\mathcal{R})$;
2. for each binary tree T on L', $|C(\mathcal{R}'_2, T) - N\binom{p}{3}/3| \leq cNp^{5/2} \log p$;
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 $C(\mathcal{R}_1) + Np^3/18 = p^3 C(\mathcal{R}) + Np^3/18$ within an additive error of $cNp^{5/2} \log p$, for some constant c. Dividing by $p^3/18$, we obtain $|\frac{18C(\mathcal{R}')}{p^3} - (18C(\mathcal{R}) + N)| \leq c' Np^{-1/2} \log p$ for some constant c'. Since $N = O(n^3)$, we can choose p polynomially bounded in terms of n such that the right member is less than $\frac{1}{2}$, implying that $\lfloor \frac{18C(\mathcal{R}')}{p^3} \rfloor = 18C(\mathcal{R}) + N$. □

Corollary 3. MaxRTC *restricted to minimally dense instances is* NP*-hard.*

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

6.2 Polynomial-time inapproximability of general MinRTI

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

Theorem 6. MinRTI *is not approximable within* $\Omega(\log n)$ *unless* P = NP.

The proof of this theorem is carried out in two steps. 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 give a measure-preserving reduction from Hitting Set to MinRTI-W (Lemma 3) and a measure-preserving reduction from MinRTI-W to MinRTI (Lemma 4). The hardness of approximation of MinRTI then follows from [6].

Due to space limitations, the proof of Lemma 3 has been omitted from this conference version of the paper. Please refer to the full version for a complete proof.

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

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

Proof. Given a weighted triplet set $\mathcal{R}$ on L, construct an unweighted triplet set $\mathcal{R}'$ on L' where the label set 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 a triplet $t = xy|z \in \mathcal{T}(L) \setminus rt(T)$. If there existed an i such that $xt_i|z \in rt(T')$ and $yt_i|z \in rt(T')$, we would obtain $xy|z \in rt(T')$, which is impossible. It follows that for each $1 \leq i \leq \mathcal{R}(t)$, one of $xt_i|z$, $yt_i|z$ is not in $\mathcal{R}(t')$, and thus the contribution of these triplets to $I(\mathcal{R}', T')$ is $\geq \mathcal{R}(t)$; in other words, setting $T = T'|L$ gives a tree such that $I(\mathcal{R}, T) \leq I(\mathcal{R}', T')$. □

7 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 case	APX-hard ([5])	$(3 - \frac{2}{n-2})$-approx. (Section 4)
dense	NP-hard ($\downarrow$)	PTAS ([16])
minimally dense	NP-hard (Section 6.1)	PTAS ($\uparrow$)
MinRTI:		
general case	Inappr. $\Omega(\log n)$ (Section 6.2)	$(n-2)$-approx. ([9] + Section 2)
dense	NP-hard ($\downarrow$)	$(n-2)$-approx. ($\uparrow$)
minimally dense	NP-hard (Section 6.1)	$(n-2)$-approx. ($\uparrow$)

Significantly, MaxRTC can be approximated within a constant ratio of 3 in polynomial time whereas MinRTI cannot be approximated within a ratio of $\Omega(\log n)$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$.

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 [5], a PTAS is unlikely. Note that both of the 3-approximation algorithms `One-Leaf-Split` from [9] 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 [9].

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?

Acknowledgments

We thank Leo van Iersel, Judith Keijsper, Steven Kelk, Kazuya Maemura, Hirotaka Ono, Kunihiko Sadakane, and Leen Stougie for helpful comments.

References

1. A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.
2. N. Ailon and N. Alon. Hardness of fully dense problems. *Information and Computation*, 205(8):1117–1129, 2007.
3. N. Alon. Ranking Tournaments. *SIAM Journal of Discrete Mathematics*, 20(1):137–142, 2006.
4. D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees: Theory and Methods in Phylogenetic Analysis.* PhD thesis, University of Canterbury, Christchurch, New Zealand, 1997.
5. J. Byrka, P. Gawrychowski, K. T. Huber, and S. Kelk. Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. Submitted, 2008.

6. U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, 1998.
7. J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Inc., 2004.
8. L. Gąsieniec, J. Jansson, A. Lingas, and A. Östlin. Inferring ordered trees from local constraints. In *Proc. of* CATS'98, volume 20(3) of *Australian Computer Science Communications*, pages 67–76. Springer-Verlag Singapore, 1998.
9. L. Gąsieniec, J. Jansson, A. Lingas, and A. Östlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, 3(2–3):183–197, 1999.
10. Y. J. He, T. N. D. Huynh, J. Jansson, and W.-K. Sung. Inferring phylogenetic relationships avoiding forbidden rooted triplets. *Journal of Bioinformatics and Computational Biology*, 4(1):59–74, 2006.
11. M. R. Henzinger, V. King, and T. Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24(1):1–13, 1999.
12. J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48(4):723–760, 2001.
13. L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, F. Hagen, and T. Boekhout. Constructing level-2 phylogenetic networks from triplets. In *Proc. of* RECOMB 2008, volume 4955 of *LNCS*, pages 450–462. Springer-Verlag, 2008.
14. L. van Iersel, S. Kelk, and M. Mnich. Uniqueness, intractability and exact algorithms: reflections on level-k phylogenetic networks. Submitted, 2008.
15. J. Jansson. On the complexity of inferring rooted evolutionary trees. In *Proc. of* GRACO 2001, volume 7 of *Electronic Notes in Discrete Mathematics*, pages 121–125. Elsevier, 2001.
16. J. Jansson, A. Lingas, and E.-M. Lundell. A triplet approach to approximations of evolutionary trees. Poster H15 presented at RECOMB 2004, 2004.
17. J. Jansson, J. H.-K. Ng, K. Sadakane, and W.-K. Sung. Rooted maximum agreement supertrees. *Algorithmica*, 43(4):293–307, 2005.
18. J. Jansson, N. B. Nguyen, and W.-K. Sung. Algorithms for combining rooted triplets into a galled phylogenetic network. *SIAM Journal on Computing*, 35(5):1098–1121, 2006.
19. J. Jansson and W.-K. Sung. Inferring a level-1 phylogenetic network from a dense set of rooted triplets. *Theoretical Computer Science*, 363(1):60–68, 2006.
20. T. Jiang, P. Kearney, and M. Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM Journal on Computing*, 30(6):1942–1961, 2001.
21. M. P. Ng and N. C. Wormald. Reconstruction of rooted trees from subtrees. *Discrete Applied Mathematics*, 69(1–2):19–31, 1996.
22. R. D. M. Page. Modified mincut supertrees. In *Proc. of* WABI 2002, volume 2452 of *LNCS*, pages 537–552. Springer-Verlag, 2002.
23. C. Semple and M. Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105(1–3):147–158, 2000.
24. S. Snir and S. Rao. Using Max Cut to enhance rooted trees consistency. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):323–333, 2006.
25. B. Y. Wu. Constructing the maximum consensus tree from rooted triples. *Journal of Combinatorial Optimization*, 8(1):29–39, 2004.