International Journal of Foundations of Computer Science

Approximating the Maximum Isomorphic Agreement Subtree is Hard

Paola Bonizzoni

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, via Bicocca degli Arcimboldi 8, 20126 Milano(Italy), bonizzoni@disco.unimib.it

Gianluca Della Vedova

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, via Bicocca degli Arcimboldi 8, 20126 Milano(Italy), dellavedova@disco.unimib.it

and

Giancarlo Mauri

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, via Bicocca degli Arcimboldi 8, 20126 Milano(Italy), mauri@disco.unimib.it Received

Revised
Communicated by

ABSTRACT

The Maximum Isomorphic Agreement Subtree (MIT) problem is one of the simplest versions of the Maximum Interval Weight Agreement Subtree method (MIWT) which is used to compare phylogenies. More precisely MIT allows to provide a subset of the species such that the exact distances between species in such subset are preserved among all evolutionary trees considered. In this paper, the approximation complexity of the MIT problem is investigated, showing that it cannot be approximated in polynomial time within factor $\log^{\delta} n$ for any $\delta > 0$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(2^{polylog\, n})$ for instances containing three trees. Moreover, we show that such result can be strengthened whenever instances of the MIT problem can contain an arbitrary number of trees, since MIT shares the same approximation lower bound of MAX CLIQUE.

Keywords: computational complexity, bioinformatics, inapproximability, evolutionary tree comparison.

1. Introduction

Evolutionary trees are trees where each leaf is labeled by a distinct element in a set S of species and where all internal nodes have degree at least three. They are frequently used by biologists to represent classifications of species. More precisely, each edge is weighted with the estimated (temporal) distance between the two species represented by its endpoints. A number of methods to infer evolutionary trees have been proposed [9, 15, 5, 13, 14, 17, 31, 4, 3, 27, 8, 34], moreover it is rather common to compare the same set of species w.r.t. different biological

sequences or different genes, hence obtaining various trees. This fact motivates the compelling need to compare different trees, in order to extract a common history. The Maximum Agreement Subtree method is a basic approach that allows to reconciliate different evolutionary trees over the same set of species: it computes a subset of the extant species about which all trees "agree". A general way to define an agreement subtree from a set $T_1, \cdots, T_k$ of S-labeled trees has been formalized in [1]. This method assumes that each edge is labeled by an interval weight (a range of time to measure the duration of the evolution process) and looks for a subset S^* of the extant species S such that:

- each edge of the subtree induced in each tree of the given set is labeled by a value belonging to the given interval,
- for each pair of extant species in S^*, the distance between them is the same in all trees.

The problem stated above is called Maximum Interval Weight Agreement Subtree (MIWT), and is a very general formulation of the problem of comparing phylogenies. In order to obtain more efficient algorithmic solutions, some restrictions have been introduced to MIWT. A first natural restriction requires that each interval reduces to a single value; such problem is called Maximum Weight Agreement Subtree (MWT). A different restriction of MIWT is the one where an agreement subtree is homeomorphic to a subtree of each tree in the instance, since it is equivalent to require all intervals to be of the form $[1, n-1]$, where n is the number of extant species considered. This problem is called Maximum Homeomorphic Agreement Subtree (MHT). Note that this problem is sometimes referred to as Maximum Agreement Subtree and is abbreviated by (MAST). A third restriction of MIWT is the one where all intervals are of the form $[1, 1]$, and is called Maximum Isomorphic Agreement Subtree (MIT), as all subtrees induced by a feasible solution must be isomorphic. The MIT problem is also a restricted case of the maximum isomorphic subgraph problem, investigated in [23]. Since MIT and MHT are the two more restricted problems among the ones we have mentioned, most of the efforts to develop efficient algorithms have been concentrated on them.

Efficient algorithms for the MHT problem for instances of two trees have been widely investigated in literature. While some heuristics have been found [16, 28], the first polynomial time algorithm has been described only in 1993 by Steel and Warnow [33]. Afterwards further improvements have appeared in literature [10, 24, 29]. To our knowledge the most efficient algorithms for the problem are due to Farach and Thorup which developed a $O(n^{3/2} \log n)$ algorithm for rooted trees of bounded degree [11, 12], to Cole and Hariharan [7, 6] for the case of rooted trees of unbounded degree, which gave a $O(n \log n)$ algorithm, and to Kao, Lam, Przytycka, Sung and Ting [25] which described a technique allowing to match the time complexity of the two previously cited algorithms also in the case of unrooted trees. The problems MHT and MIT over a set of trees, where at least one of the trees has bounded degree, can be solved in polynomial time [1], even though the time complexity is exponential in the bound for the degree. Moreover both problems

are **NP**-hard for instances containing three trees of unbounded degree, hence it is necessary to focus on designing polynomial time approximation algorithms. The approximation complexity of the MHT problem has been deeply investigated in [21], where some strong negative results have been obtained. Since the MIT is a different restriction of the MIWT, it seems natural to investigate if the negative results for MHT hold also for MIT or the latter problem is easier to approximate than the former one. In our paper we show that the negative results of [21] hold also for the MIT problem, as a consequence of a nontrivial application of the self-improvement technique. Applying self-improvement usually leads to a result of the form "either problem Π admits a PTAS or Π cannot be approximated within a constant factor unless **NP**=**P** (or another unlikely collapse between complexity classes occur). This idea has been exploited in [18] to prove that Max Independent Set either has a PTAS or no constant factor polynomial-time approximation algorithm (unless **NP**=**P**) and has been successively pushed further by Karger et. al. in [26] to prove that the Longest Path cannot be approximated within $O(logn)$ unless **P**=**NP**. In the latter paper the inapproximability result has been obtained by combining the self-improvement technique and an L-reduction (to prove that the problem is **MAX SNP**-hard). Consequently the "easy" way to prove that MIT is hard to approximate seems to rely on the **MAX SNP**-hardness proof by Amir and Keselman [1] and on the application of the self-improvement technique to the problem. Unfortunately the MIT problem seems to lack some of the properties that in [21] have been exploited implicitly in the proofs. Hence in our paper we deal with a restriction of MIT, called R-MIT, that has the desired properties, but before applying self-improvement to R-MIT we have to prove that the latter problem is **MAX SNP**-hard.

As a consequence of our results achieving a constant error ratio is an **NP**-hard problem, even for instances consisting of only three trees. Moreover we have strengthened such negative results in the case of instances containing an arbitrary number of trees (in the restricted case where each tree in the instance has depth 2), as we show that MIT shares exactly the same inapproximability properties of MAX CLIQUE [20], implying that there cannot exist a polynomial time $n^{1-\epsilon}$ ratio approximation algorithm for each $\epsilon > 0$, unless **NP**=**ZPP**. A similar, but slightly weaker, negative result on the approximability of MHT has been obtained in [19], showing that such problem cannot be approximated within factor n^{ϵ} for any $0 \leq \epsilon < \frac{1}{9}$, since approximating MHT within factor n^{ϵ} in polynomial time implies a polynomial-time approximation algorithm for MAX CLIQUE with $n^{3\epsilon+o(1)}$ guaranteed approximation ratio.

2. Preliminaries

All trees we will deal with in this paper are rooted, that is we distinguish a special vertex of the tree T and we call such a vertex *root*, denoted by $r(T)$. All results presented in the paper are referred to rooted trees, but they can be generalized to the unrooted case.

Let $S = \{s_1, \ldots, s_n\}$ be a set of labels. An S-labeled tree has n leaves, each one labeled with a distinct element of S; since each label identifies unambiguously

a leaf of the tree, in the following of the paper we will write a label x meaning the leaf of the tree with label x. The Maximum Isomorphic Agreement Subtree Problem (shortly MIT) is defined formally as follows:
Instance: a set $\mathcal{T} = \{T_1, \ldots, T_m\}$ of S-labeled trees.
Solution: an S^*-labeled tree T^*, with $S^* \subseteq S$, such that T^* is isomorphic to a subtree of all trees in $\mathcal{T}$.
Measure: $|S^*|$, to be maximized.

The Maximum Homeomorphic Agreement Subtree (shortly MHT) is:
Instance: a set $\mathcal{T} = \{T_1, \ldots, T_m\}$ of S-labeled trees.
Solution: an S^*-labeled tree T^*, with $S^* \subseteq S$, such that T^* does not contain any internal node (with the possible exception of the root) of degree 2 and, for each tree $T_i \in \mathcal{T}$, T^* is homeomorphic to a subtree of T_i.
Measure: $|S^*|$, to be maximized.

Let T be a tree and let a, b be two nodes of T, then we will denote by $d_T(a, b)$ the distance between a and b in T, that is the number of edges in the unique simple path from a to b in T. Let T be a rooted tree, and let t be a node of T, then the depth of t in T is the distance of t from the root of T. The depth of a tree T, denoted by depth(T), is the maximum among the depths of its nodes. Given two leaves a, b of T we define the *least common ancestor*, of a and b in T, denoted by $\mathrm{lca}_T(a, b)$, as the maximum depth node of T which is ancestor of both a and b.

It is immediate to note that the **NP**-completeness proof given by Amir and Keselman in [1] is an L-reduction for the MHT problem, as pointed out in [21]. Similarly it is possible to prove that MIT is **MAX SNP**-hard, that is there is no polynomial time approximation scheme for it, unless **P**=**NP**.

Anyway, differently from [21], we have to deal with a restricted version of the problem in order to prove our inapproximability results, hence such **MAX SNP**-hardness proof is not adequate to our purposes. More precisely we consider only instances consisting of trees having leaves all at the same depth in every tree. Formally $d_{T_i}(a, r(T_i)) = d_{T_j}(b, r(T_j))$ for all $a, b \in S$ and every pair of trees T_i, T_j in the instance. We will say that trees in such instances are *restricted*. This new problem will be called R-MIT. Clearly all inapproximability results for this problem hold also for MIT.

The following Lemma, proved in [32], characterizes all feasible solutions of each instance of R-MIT.

Lemma 2.1 *Let $\mathcal{T}$ be a set of S-labeled trees, and let $S^* \subseteq S$. Then there exists a S^*-labeled tree T^* that is isomorphic to a subtree of each tree in $\mathcal{T}$ if and only if for each pair of labels $a, b \in S^*$, a and b have the same distance in all trees in $\mathcal{T}$.*

As a consequence we can identify a feasible solution of an instance of MIT as a subset of its label set. The following property of trees, whose straightforward proof is omitted, will be used in the remaining of the paper.

Proposition 2.2 *Let a, b be two leaves of a S-labeled tree with root r. Then $d_T(a, b) = d_T(a, r) + d_T(b, r) - 2\, d_T(r, lca_T(a, b))$.*

3. R-MIT is MAX SNP-hard

In this section we are going to prove that the R-MIT problem is **MAX SNP**-hard. This results is necessary to prove that MIT is hard to approximate even on instances consisting of only three trees.

The first step is to rule out the possibility of having a PTAS, that is a polynomial time $(1+\epsilon)$-approximation algorithm for all fixed constants $\epsilon > 0$, for R-MIT, by describing an L-reduction (the definition of L-reduction among optimization problems has been given in [30]).

The problem used in the L-reduction to R-MIT is the Tridimensional Bounded Matching (shortly 3DM-B), formally defined as follows:
Instance: three pairwise disjoint sets $< X_1, X_2, X_3 >$ and a set M of distinct triples where $M \subseteq X_1 \times X_2 \times X_3$ and every element in $X_1 \cup X_2 \cup X_3$ occurs in at least one and at most B triples of M.
Solution: a subset M_1 of M, such that no two triples in M_1 share a common element.
Measure: $|M_1|$, to be maximized.
The general 3DM-B problem is **MAX SNP**-hard [22].

Let $\mathcal{M} =< X_1, X_2, X_3, M >$ be an instance of the 3DM-B problem, with $M \subseteq X_1 \times X_2 \times X_3$, $X_i = \{x_{i,1}, x_{i,2}, \ldots x_{i,|X_i|}\}$. Then we will associate to $\mathcal{M}$ an instance $< T_1, T_2, T_3 >$ of MIT. Each tree T_i consists of the following nodes and edges: a root labeled r_i, a node connected to the root for each element $x_{i,j}$ of X_i, and each triple $(x_{1,j}, x_{2,j}, x_{3,j}) \in \mathcal{M}$ is a leaf of T_i connected to $x_{i,j}$. Consequently each tree T_i is M-labeled.

In Fig. 1 it is represented the instance of R-MIT associated to the instance of 3DM-B where $X_1 = \{x_{1,1}, x_{1,2}, x_{1,3}\}$, $X_2 = \{a, c\}$, $X_3 = \{b, d\}$ and $M = \{(x_{1,1}ab), (x_{1,1}cd), (x_{1,2}cd), (x_{1,3}cd)\}$.

Fig. 1. Example of instance of R-MIT associated to an instance of 3DM-B

Since the distance from each node to the root is 2 in all trees of the instance of MIT associated to an instance of 3DM-B, such set of trees is an instance of R-MIT. The following Lemma is an immediate consequence of such fact.

Lemma 3.1 *Let $\mathcal{M} =< X_1, X_2, X_3, M >$ be an instance of* 3DM-B*, and let $< T_1, T_2, T_3 >$ be the associated instance of* MIT*. Given a tree T_i with $1 \leq i \leq 3$, and given two distinct leaves s, t of T_i, then the distance between s and t in T_i is 2 or 4.*

Note that the distance of two leaves s and t in a tree T_i is 2 if and only if s and t are labeled by triples of M that share the same element in the set X_i.

Lemma 3.2 *Let $\mathcal{M} =< X_1, X_2, X_3, M >$ be an instance of* 3DM-B*, let $S \subseteq M$, and let $< T_1, T_2, T_3 >$ be the instance of* R-MIT *associated to $\mathcal{M}$. Then S is a feasible solution of $< T_1, T_2, T_3 >$ if and only if each pair s, t of distinct triples in S has distance 4 in all trees T_i.*

Proof. By Lemma 2.1 S is a feasible solution if and only if each distinct pair s, t of triples in S have the same distance in all trees T_i, which is, by Lemma 3.1 either 2 or 4. Let us assume that there exists a pair s, t that has distance 2 in all trees. Then by construction s is equal to t, contradicting the fact that all triples in M are distinct, hence for each pair the distance must be 4, as stated. The other direction follows immediately by Lemma 2.1. □

From Lemmas 3.1 and 3.2 the reduction from 3DM-B to R-MIT that we have described can be thought as a polynomial-time computable function r that associates to each instance $\mathcal{M}$ of 3DM-B an instance $r(\mathcal{M})$ of R-MIT and a polynomial-time computable function s that associates to each feasible solution Apx of $r(\mathcal{M})$ a feasible solution $s(Apx)$ such that the costs of Apx and of $s(Apx)$ are the same and the optima of $\mathcal{M}$ and of $r(\mathcal{M})$ are the same. This implies that our reduction is an L-reduction, hence R-MIT is **MAX SNP**-hard. The following theorem follows from the results by Arora et. al. given in [2].

Theorem 3.3 *There does not exists a PTAS for* R-MIT *unless* **P**=**NP**.

4. Product of trees

The inapproximability result over instances of three trees is obtained by means of the *self-improvement* technique. In [21] such technique has been exploited to prove a similar result for the MHT problem. Such technique requires a careful definition of a product between instances of the problem, which is defined as follows:

Definition 4.1 *Let T_1 be a S_1-labeled tree, T_2 a S_2-labeled tree and, for a given leaf s of T_1, $T_{2 \cdot s}$ is the tree obtained from T_2 relabeling each leaf s_2 with the sequence ss_2. Then the product $T_1 \cdot T_2$ is the tree obtained from T_1 replacing each leaf s with the tree $T_{2 \cdot s}$.*

Let T be a S-labeled tree, then $T^2 = T \cdot T$ and $T^i = T^{i-1} \cdot T$, $i > 2$. Note that the label of a leaf of the tree T^k is a string $s_1 \dots s_k$ of k symbols over the alphabet S. An immediate property of the product of trees is stated below:

Proposition 4.1 *Let T_1, T_2 be two restricted trees. Then $T_1 \cdot T_2$ is also a restricted tree.*

The following Lemma points out the motivation for our definition of product.

Lemma 4.2 *Let T_1, T_2 be two restricted S-labeled trees, let a, b be two labels in S and let α, β be two strings of $k-1$ symbols over S. Then $d_{T_1^k}(\alpha a, \beta b) = d_{T_2^k}(\alpha a, \beta b)$ if and only if $d_{T_1}(a, b) = d_{T_2}(a, b)$ and $d_{T_1^{k-1}}(\alpha, \beta) = d_{T_2^{k-1}}(\alpha, \beta)$*

Proof. Since T_1 and T_2 are restricted trees (that is in T_1 and T_2 all leaves have the same depth), and by Proposition 2.2, in order to prove the lemma, it is sufficient to show that $d_{T_1^k}(\mathrm{lca}_{T_1^k}(\alpha a, \beta b), r(T_1^k)) = d_{T_2^k}(\mathrm{lca}_{T_2^k}(\alpha a, \beta b), r(T_2^k))$ if and only

Fig. 2. Example of instance of R-MIT associated to an instance of 3DM-B

if $d_{T_1}(\mathrm{lca}_{T_1}(a,b), r(T_1)) = d_{T_2}(\mathrm{lca}_{T_2}((a,b), r(T_2))$ and $d_{T_1^{k-1}}(\mathrm{lca}_{T_1^{k-1}}(\alpha,\beta), r(T_1^{k-1}))$ $= d_{T_2^{k-1}}(\mathrm{lca}_{T_2^{k-1}}(\alpha,\beta), r(T_2^{k-1}))$. Assume initially that $\alpha = \beta$, then, by definition of product, $d_{T_1^k}(\mathrm{lca}_{T_1^k}(\alpha a, \beta b), r(T_1^k)) = d_{T_{1\cdot\alpha}}(\mathrm{lca}_{T_{1\cdot\alpha}}(\alpha a, \alpha b),\ r(T_{1\cdot\alpha})) + \mathrm{depth}(T_1^{k-1})$ and $d_{T_2^k}(\mathrm{lca}_{T_2^k}(\alpha a, \beta b), r(T_2^k)) = d_{T_{2\cdot\alpha}}(\mathrm{lca}_{T_{2\cdot\alpha}}(\alpha a, \alpha b), r(T_{2\cdot\alpha}))+\mathrm{depth}(T_2^{k-1})$, hence the two distances $d_{T_1^k}(\mathrm{lca}_{T_1^k}(\alpha a, \beta b), r(T_1^k))$ and $d_{T_2^k}(\mathrm{lca}_{T_2^k}(\alpha a, \beta b), r(T_2^k))$ are the same if and only if $d_{T_1}(\mathrm{lca}_{T_1}(a,b), r(T_1)) = d_{T_2}(\mathrm{lca}_{T_2}((a,b), r(T_2))$. Assume now that $\alpha \neq \beta$, then $\mathrm{lca}_{T_1^k}(\alpha a, \beta b)) = \mathrm{lca}_{T_1^{k-1}}(\alpha,\beta)$ and $\mathrm{lca}_{T_2^k}(\alpha a, \beta b) = \mathrm{lca}_{T_2^{k-1}}(\alpha,\beta)$. As a consequence $d_{T_1^k}(\mathrm{lca}_{T_1^k}(\alpha a, \beta b), r(T_1^k)) = d_{T_2^k}(\mathrm{lca}_{T_2^k}(\alpha a, \beta b), r(T_2^k))$ if and only if $d_{T_1^{k-1}}(\mathrm{lca}_{T_1^{k-1}}(\alpha,\beta), r(T_1^{k-1}))$ is equal to $d_{T_2^{k-1}}(\mathrm{lca}_{T_2^{k-1}}(\alpha,\beta), r(T_2^{k-1}))$. This suffices to prove the Lemma. □

The following lemma relates a feasible solution of $< T_1, T_2, T_3 >$ with a feasible solution of $< T_1^k, T_2^k, T_3^k >$.

Lemma 4.3 *Let $< T_1, T_2, T_3 >$ be an instance of* R-MIT*, and let F be a feasible solution with cost $cost(F)$ of such an instance. Then it is possible to compute in polynomial time a solution of $< T_1^k, T_2^k, T_3^k >$ whose cost is $cost(F)^k$.*

Proof. Let F_k be the set of strings of labels $\{f_1 \cdots f_k : f_i \in F,\ 1 \leq i \leq k\}$, then for each pair of strings of labels $f_{\alpha_1} \cdots f_{\alpha_k},\ f_{\beta_1} \cdots f_{\beta_k}$ in F_k, their distance is the same in all trees T_1^k, T_2^k, T_3^k, since for each $1 \leq i \leq k$ $d_{T_1}(f_{\alpha_i}, f_{\beta_i}) = d_{T_2}(f_{\alpha_i}, f_{\beta_i}) = d_{T_3}(f_{\alpha_i}, f_{\beta_i})$, as all $f_{\alpha_i}, f_{\beta_i}$ are in S and from Prop. 2.2. □

Lemma 4.4 *Let $< T_1^k, T_2^k, T_3^k >$ be an instance of* R-MIT *and let S_k be a feasible solution of $< T_1^k, T_2^k, T_3^k >$, then it is possible to compute in polynomial time a feasible solution S_1 of $< T_1, T_2, T_3 >$ such that $cost(S_k) \leq cost(S_1)^k$.*

Proof. Let $S_k = \{f_{\alpha_1}, \ldots, f_{\alpha_k}\}$ be a feasible solution of $< T_1^k, T_2^k, T_3^k >$. By applying Lemma 4.2 iteratively we can obtain k feasible solutions F_i of $< T_1, T_2, T_3 >$, where each solution F_i contains exactly the symbols f_{α_i} of S that are in the i-th position of a string in S_k. Let F^* be the largest such F_i and let F_k^* be the set of strings $\{f_1 \ldots f_k : f_j \in F^*,\ 1 \leq j \leq k\}$. Just as in the proof of Lemma 4.3 it is possible to prove that F_k^* is a feasible solution of $< T_1^k, T_2^k, T_3^k >$. An immediate

counting argument and the fact that F^* is the F_i of maximum cardinality imply that $|S_k| \leq |F_k^*| = |F^*|^k$. □

In the following, given an instance $< T_1, T_2, T_3 >$ of R-MIT and an approximation algorithm for R-MIT, we will denote by $\text{apx}(< T_1, T_2, T_3 >)$ the solution returned by such algorithm for the instance $< T_1, T_2, T_3 >$, while $\text{Opt}(< T_1, T_2, T_3 >)$ denotes the optimum solution. The basic idea of the proofs of the main results in this section is sketched in the following: given an instance $< T_1, T_2, T_3 >$ of R-MIT we expand it (by Lemma 4.3) to another instance $< T_1^k, T_2^k, T_3^k >$ to which we apply an hypothetical approximation algorithm. By Lemma 4.4 we are able to infer from the approximate solution of $< T_1^k, T_2^k, T_3^k >$ an approximate solution of $\text{apx}(< T_1, T_2, T_3 >)$ whose approximation factor is "much better" than the one we have got for $\text{apx}(< T_1^k, T_2^k, T_3^k >)$.

We now state our main results, where all logarithms have natural bases.

Theorem 4.5 *There does not exists a polynomial-time constant-ratio approximation algorithm for* R-MIT *unless* **NP**=**P**.

Proof. Assume that there exists a ϵ-approximation algorithm with polynomial time complexity for R-MIT. Pose $k = \lceil \log \epsilon \rceil$, consequently $e^k \geq \epsilon$. Then, by Lemmas 4.3, 4.4,

$$\left(\frac{\text{Opt}(< T_1, T_2, T_3 >)}{\text{apx}(< T_1, T_2, T_3 >)} \right)^k = \frac{\text{Opt}(< T_1^k, T_2^k, T_3^k >)}{\text{apx}(< T_1^k, T_2^k, T_3^k >)} \leq \epsilon \leq e^k$$

hence $(\frac{\text{Opt}(<T_1,T_2,T_3>)}{\text{apx}(<T_1,T_2,T_3>)}) \leq e$. Please note that computing $< T_1^k, T_2^k, T_3^k >$ from $< T_1, T_2, T_3 >$ can be done in $O(n^{\lceil \log \epsilon \rceil})$ time, hence we have described a PTAS for R-MIT. By Theorem 3.3 **NP**=**P**. □

Corollary 4.6 *There exists $\delta > 0$ such that* R-MIT *cannot be approximated within factor $\log^\delta n$ in polynomial time, unless* **NP**$\subseteq$**DTIME**$[2^{polylog\, n}]$.

Proof. Assume that for all $\delta > 0$ there exists a $\log^\delta n$-approximation polynomial-time algorithm for R-MIT, we will prove that there exists an e-approximation polynomial-time algorithm. Pose $k = \lceil \log(\log^\delta n) \rceil$, consequently $e^k \geq \log^\delta n$. Just as in the proof of Theorem 4.5 we will denote with $\text{apx}(< T_1, T_2, T_3 >)$ the solution returned by the approximation algorithm for the instance $< T_1, T_2, T_3 >$, while $\text{Opt}(< T_1, T_2, T_3 >)$ denotes the optimum solution. Then, by Lemmas 4.3, 4.4,

$$\left(\frac{\text{Opt}(< T_1, T_2, T_3 >)}{\text{apx}(< T_1, T_2, T_3 >)} \right)^k = \frac{\text{Opt}(< T_1^k, T_2^k, T_3^k >)}{\text{apx}(< T_1^k, T_2^k, T_3^k >)} \leq \log^\delta n$$

taking the logarithms of both sides

$$k \log \left(\frac{\text{Opt}(< T_1, T_2, T_3 >)}{\text{apx}(< T_1, T_2, T_3 >)} \right) \leq \log(\log^\delta n)$$

Consequently

$$\lceil \log(\log^\delta n) \rceil \log \left(\frac{\text{Opt}(< T_1, T_2, T_3 >)}{\text{apx}(< T_1, T_2, T_3 >)} \right) \leq \log(\log^\delta n)$$

implying that $\log(\frac{\text{Opt}(<T_1,T_2,T_3>)}{\text{apx}(<T_1,T_2,T_3>)}) \leq 1$. Hence $\frac{\text{Opt}(<T_1,T_2,T_3>)}{\text{apx}(<T_1,T_2,T_3>)} \leq e$. It is immediate to note that that computing $< T_1^k, T_2^k, T_3^k >$ from $< T_1, T_2, T_3 >$ can be done in $O(n^{\lceil \log \log^\delta n \rceil}) = 2^{polylog n}$ time. Thus the claim follows from Thm. 4.5. □

5. Inapproximability over unbounded number of trees

The inapproximability result presented in the previous section can be strengthened when instances are not required to contain exactly three trees, but can contain an arbitrary number of trees (even in the case of all trees of depth 2). This can be proved by a simple L-reduction from MAX CLIQUE. Since such reduction preserves the optimum and the cost of approximate solutions, MIT with unbounded number of trees inherits the same inapproximability results of MAX CLIQUE, which cannot be approximated within $n^{1-\epsilon}$ for each $\epsilon > 0$, unless **ZPP=NP**. [20]. The formal definition of the MAX CLIQUE problem follows:
Instance: an unoriented graph $G =< V, E >$.
Solution: a *clique* of G, that is is a subset $C \subseteq V$ such that $(c_1, c_2) \in E$ for each pair c_1, c_2 of vertices in C.
Measure: $|C|$, to be maximized.

The reduction is quite simple: let $G = (V, E)$ be a graph with $E \neq \emptyset$. The instance of MIT contains the V-labeled trees in the set $\{T_{edge}\} \cup \{T_{ij} : i, j \in V, (i, j) \notin E\}$, where T_{edge} has root r and each leaf v of T_{edge} has p_v as parent and p_v is a child of r. Each tree T_{ij} consists of a root r, a node p_{ij} that is the parent of both leaves v_i, v_j a node p_z for each $z \in V - \{v_i, v_j\}$ and each p_z is the parent of the leaf v_z. Moreover p_{ij} and all p_z with $z \in V - \{i, j\}$ are the children of the root.

Fig. 3. Example of reduction from MAX CLIQUE.

An example of application of such reduction is represented in Fig. 3. The following Lemma points out the structure of all feasible solutions considered in our reduction.

Lemma 5.1 *Let $G =< V, E >$ be an instance of* MAX CLIQUE*, let $\mathcal{T}$ be the set of V-labeled trees associated to G, and let T be a feasible solution of* MIT($\mathcal{T}$)*. Let*

v_1, v_2 be two distinct leaves of T. Then the distance between v_1 and v_2 in T is four.

Proof. Let v_1, v_2 be two distinct leaves of T. Since v_1 and v_2 are both in a feasible solution T of $\mathcal{T}$, by Lemma 2.1 their distance must be the same in all trees in $\mathcal{T}$. Since $d_{T_{edge}}(v_1, v_2) = 4$ then $d_T(v_1, v_2) = 4$. □

We will show how to compute a feasible solution of MIT from a feasible solution of MAX CLIQUE and vice versa, so that the costs of both solutions are the same.

Let $\mathcal{T}$ be the instance of MIT, associated to the instance $G =< V, E >$ of MAX CLIQUE, and let $V_1 \subset V$ be a feasible solution of $\mathcal{T}$. Please note that, by Lemmas 2.1 and 5.1 a subset $V_1 \subseteq V$ is a feasible solution of $\mathcal{T}$ if and only if $d_T(v_1, v_2) = 4$ for each pair of distinct elements $v_i, v_j \in V_1$ and each tree $T \in \mathcal{T}$. We will prove that V_1 is a clique of G. Assume to the contrary that V_1 is not a clique of G, that is there exist two vertices $v_i, v_j \in V_1$ such that $(v_i, v_j) \notin E$. By construction in $\mathcal{T}$ there is the tree T_{ij}, and $d_{T_{ij}}(v_i, v_j) = 2$. Consequently by Lemma 5.1 v_i and v_j cannot both be in a feasible solution of $\mathcal{T}$. To compute a feasible solution of $\mathcal{T}$ from a clique of G is trivial, hence the following theorem follows:

Theorem 5.2 MIT *over an unbounded number of trees cannot be approximated within $n^{1-\epsilon}$ for each $\epsilon > 0$, unless* **ZPP**=**NP**.

6. Conclusions

The MIT problem is one of the simplest formulations of evolutionary trees comparison proposed in literature, while the most studied of such formulations is the MHT problem. In our paper we have shown that MIT shares the same inapproximability bounds of MHT whenever the instances are restricted to contain exactly 3 trees, while it inherits the same bounds of MAX CLIQUE when the instances are unrestricted.

Acknowledgments

We thank the anonymous referees which have helped in clarifying the presentation of the paper. We gratefully acknowledge the support of MURST grant "Bioinformatica e Ricerca genomica".

References

1. A. Amir and D. Keselman. Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms. *SIAM Journal on Computing*, **26(6)**:1656–1669, 1997.
2. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of ACM*, **45(3)**:501–555, 1998.
3. A. Ben-Dor, B. Chor, D. Graur, R. Ophir, and D. Pellep. From four-taxon trees to phylogenies: The case of mammalian evolution. In *Proceedings of the 2nd Annual International Conference on Computational Molecular Biology*, pages 9–19, 1998.
4. V. Berry and O. Gascuel. Inferring evolutionary trees with strong combinatorial evidence. In *Proceedings of the 3rd International Computing and Combinatorics Conference*, pages 111–120, 1997.

5. P. Buneman. The recovery of trees from measures of dissimilarity. In Hodson, Kendall, and Tautu, editors, *Mathematics in the Archaeological and Historical Sciences*. Edinburgh University Press, 1971.
6. R. Cole, M. Farach, R. Hariharan, T. Przytycka, and M. Thorup. An O($n \log n$) algorithm for the maximum agreement subtree problem for binary trees. *SIAM Journal on Computing*, to appear.
7. R. Cole and R. Hariharan. An O($n \log n$) algorithm for the maximum agreement subtree problem for binary trees. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA96)*, pages 323–332, 1996.
8. P. Erdös, M. Steel, L. Székely, and T. Warnow. Constructing big trees from short sequences. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, pages 827–837, 1997.
9. P. Erdös, M. Steel, L. A. Székely, and T. Warnow. Constructing big trees from short sequences. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium*, volume 1256 of *Lecture Notes in Computer Science*, pages 827–837, 1997.
10. M. Farach, T. M. Przytycka, and M. Thorup. On the agreement of many trees. *Information Processing Letters*, **55(6)**:297–301, 1995.
11. M. Farach and M. Thorup. Fast comparison of evolutionary trees. *Information and Computation*, **123(1)**:29–37, 1995.
12. M. Farach and M. Thorup. Sparse dynamic programming for evolutionary-tree comparison. *SIAM Journal on Computing*, **26(1)**:210–230, 1997.
13. J. Felsenstein. Cases in which parsimony and compatibility will be positively misleading. *Systematic Zoology*, **27**:401–410, 1978.
14. J. Felsenstein. Evolutionary trees from DNA sequences: A maximal likelihood approach. *Journal of Molecular Evolution*, **17**:368–386, 1981.
15. J. Felsenstein. Numerical methods for inferring evolutionary trees. *Quart. Rev. Biol.*, pages 379–404, 1982.
16. C. Finden and A. Gordon. Obtaining common pruned trees. *Journal of Classification*, **2**:255–276, 1985.
17. W. M. Fitch. Toward defining the course of evolution: Minimal change for a specific tree topology. *Systematic zoology*, **20**:406–441, 1971.
18. M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.
19. L. Gasieniec, J. Jansson, A. Lingas, and A. Östlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, **3(2/3)**:183–197, 1999.
20. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, to appear.
21. J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, **71**:153–169, 1996.
22. V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, **37(1)**:27–35, 1991.
23. V. Kann. On the approximability of the maximum common subgraph problem. In *Proc. 9th Ann. Symp. on Theoretical Aspects of Comput. Sci. (STACS92)*, volume 577 of *LNCS*, pages 377–388, 1992.
24. M.-Y. Kao. Tree contractions and evolutionary trees. *SIAM Journal on Computing*,

to appear.

25. M.-Y. Kao, T. W. Lam, T. M. Przytycka, W.-K. Sung, and H.-F. Ting. General techniques for comparing unrooted evolutionary trees. In *Proceedings of the 29th Symposium on the Theory of Computing (STOC97)*, pages 54–65, 1997.
26. D. Karger, R. Motwani, and G. Ramkumar. On approximating the longest path in a graph. *Algorithmica*, **18(1)**:82–98, 1997.
27. P. Kearney. The ordinal quartet method. In *Proceedings of the 2nd Annual International Conference on Computational Molecular Biology*, pages 125–134, 1998.
28. E. Kubicka, G. Kubicki, and F. McMorris. An algorithm to find agreement subtrees. *Journal of Classification*, **12(1)**:91–99, 1995.
29. T. Lam, W. Sung, and H. Ting. Computing the unrooted maximum agreement subtree in subquadratic time. In *Proc. of the 5th Scandinavian Workshop on Algorithms Theory*, LNCS, pages 124–135, 1996.
30. C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, **43**:425–440, 1991.
31. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, **4**:406–425, 1987.
32. E. Smolenskii. *Jurnal Vicisl. Mat. i Matem. Fiz.*, **2(2)**:371–372, 1962.
33. M. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, **48(2)**:77–82, 1993.
34. K. Strimmer and A. von Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Molecular Biology and Evolution*, **13(7)**:964–969, 1996.