LCNS, Vol 2089, pp. 241–256, Springer 2001

Minimum Quartet Inconsistency is Fixed Parameter Tractable

Jens Gramm* Rolf Niedermeier

Wilhelm-Schickard-Institut für Informatik, Universität Tübingen,
Sand 13, D-72076 Tübingen, Fed. Rep. of Germany
gramm,niedermr@informatik.uni-tuebingen.de

Abstract. We study the parameterized complexity of the problem to reconstruct a binary (evolutionary) tree from a complete set of quartet topologies in the case of a limited number of errors. More precisely, we are given n taxa, exactly one topology for every subset of 4 taxa, and a positive integer k (the parameter). Then, the *Minimum Quartet Inconsistency* (MQI) problem is the question of whether we can find an evolutionary tree inducing a set of quartet topologies that differs from the given set in only k quartet topologies. MQI is NP-complete. However, we can compute the required tree in worst case time $O(4^k \cdot n + n^4)$—the problem is fixed parameter tractable. Our experimental results show that in practice, also based on heuristic improvements proposed by us, even a much smaller exponential growth can be achieved. We extend the fixed parameter tractability result to weighted versions of the problem. In particular, our algorithm can produce *all* solutions that resolve at most k errors.

1 Introduction

In recent years, quartet methods for reconstructing evolutionary trees have received considerable attention in the computational biology community [6, 11]. In comparison with other phylogenetic methods, an advantage of quartet methods is, e.g., that they can overcome the data disparity problem (see [6] for details). The approach is based on the fact that an evolutionary tree is uniquely characterized by its set of induced quartet topologies [5]. Herein, we consider an *evolutionary tree* to be an unrooted binary tree T in which the leaves are bijectively labeled by a set of taxa S. A quartet, then, is a size four subset $\{a, b, c, d\}$ of S, and the topology for $\{a, b, c, d\}$ induced by T simply is the four leaf subtree of T induced by $\{a, b, c, d\}$. The three possible quartet topologies for $\{a, b, c, d\}$ are $[ab|cd]$, $[ac|bd]$, and $[ad|bc]$.[1] E.g., the topology is $[ab|cd]$ when, in T, the paths from a to b and from c to d are disjoint. The fundamental goal of quartet

* Work supported by the DFG projects "KOMET," LA 618/3-3, and "OPAL" (optimal solutions for hard problems in computational biology), NI-369/2-1.

[1] The fourth possible topology would be the star topology, which is not considered here because it is not binary.

methods is, given a set of quartet topologies, to reconstruct the corresponding evolutionary tree. The computational interest in this paradigm derives from the fact that the given set of quartet topologies usually is fault-prone.

In this paper, we focus on the following, perhaps most often studied optimization problem in the context of quartet methods.

> Minimum Quartet Inconsistency (MQI)
> **Input:** A set S of n taxa and a set Q_S of quartet topologies such that there is exactly one topology for *every* quartet set[2] corresponding to S and a positive integer k.
> **Question:** Is there an evolutionary tree T where the leaves are bijectively labeled by the elements from S such that the set of quartet topologies induced by T differs from Q_S in at most k quartet topologies?

MQI is NP-complete [12]. Concerning the approximability of MQI, it is known that it is polynomial time approximable with a factor n^2 [11,12]. It is an open question of [11] whether MQI can be approximated with a factor at most n or even with a constant factor. The parameterized complexity [7] of MQI, however, so far, has apparently been neglected—we close this gap here. Assuming that the number k of "wrong" quartet topologies is small in comparison with the total number of given quartet topologies, we show that MQI is *fixed parameter tractable*; that is, MQI can be solved exactly in worst case time $O(4^k n + n^4)$. Observe that the input size is $O(n^4)$. It is worth noting here that the variant of MQI where the set Q_S is not required to contain a topology for every quartet is NP-complete, even if $k = 0$ [16]. Hence, this excludes parameterized complexity studies and also implies inapproximability (with any factor).

To develop our algorithm, we exhibit some nice combinatorial properties of MQI. For instance, we point out that "global conflicts" due to erroneous quartet topologies can be reduced to "local conflicts." The basis for this was laid by Bandelt and Dress [2]. This is the basic observation in order to show fixed parameter tractability of MQI. Our approach makes it possible to construct *all* evolutionary trees that can be (uniquely) obtained from the given input by changing at most k quartet topologies. This puts the user of the algorithm in the position to pick (e.g., based on additional biological knowledge) the probably best, most reasonable solution or to construct a consensus tree from all solutions. Moreover, our method also generalizes to weighted quartets.

We performed several experiments on artificial and real (fungi) data and, thereby, showed that our algorithm (due to several tuning tricks) in practice runs much faster than its theoretical (worst case) analysis predicts. For instance, with a small k (e.g., $k = 100$), we can solve relatively large ($n = 50$ taxa) instances optimally in around 40 minutes on a LINUX PC with a Pentium III 750 MHz processor and 192 MB main memory.

A full version (containing all proofs) is available [10].

[2] Note that given n species, there are $\binom{n}{4} = O(n^4)$ corresponding quartet topologies.

2 Preliminaries

Minimum quartet inconsistency. In order to find the "best" binary tree for a given set of quartet topologies, we can ask for a tree that violates a minimum number of topologies. In case we are given exactly one quartet topology for every set of four taxa, this question gives the MQI problem. If there is not a quartet topology for necessarily every set of four taxa, Ben-Dor *et al.* [3] propose two solutions, namely, a heuristic approach and an exact algorithm. The heuristic solution is based on semidefinite programming and does not guarantee to produce the optimal solution, but has a polynomial running time. The exact algorithm uses dynamic programming for finding the optimal solution and has exponential running time, namely, $O(m3^n)$, where n is the number of species and m is the number of given quartet topologies. Note that Ben-Dor *et al.* run all their experiments on MQI instances, i.e., there was exactly one quartet topology for every set of four taxa. In that case, we have $m = O(n^4)$. The memory requirement of their exact solution is $\Theta(2^n)$. According to Jiang *et al.* [11] there is a factor n^2-approximation, and, at the same time, they asked about better approximation results. Note that the complement problem of MQI, where one tries to maximize $|Q_T \cap Q|$ (Q_T being the set of quartet topologies induced by a tree T), possesses a polynomial time approximation scheme [12].
Some notation. Assume that we are given a set of n taxa S. For a quartet $\{a, b, c, d\} \subseteq S$, we refer to its possible quartet topologies by $[ab|cd]$, $[ac|bd]$, and $[ad|bc]$. These are the only possible topologies up to isomorphism. A set of quartet topologies is *complete* if it contains exactly one topology for every quartet of S. A complete set of quartet topologies over S we denote by Q_S. A set of quartet topologies Q is *tree-consistent* [2] if there exists a tree T such that for the set Q_T of quartet topologies induced by T, we have $Q \subseteq Q_T$. Set Q is *tree-like* [2] if there exists a tree with $Q = Q_T$. Since an evolutionary tree is uniquely characterized by the topologies for all its quartets [5], a complete set of topologies is tree-consistent iff it is tree-like. A set of topologies has a "conflict" whenever it is not tree-consistent. We will call a conflict "global" when a complete set of topologies is not tree-consistent. We call it "local" when a size three set of topologies, which necessarily is incomplete, is not tree-consistent.

3 Global conflicts are local

Given a complete set of quartet topologies which is not tree-consistent, the results of Bandelt and Dress [2] imply that there already is a subset of only three quartet topologies which is not tree-consistent. This is the key to developing a fixed parameter solution for the problem: It is sufficient to examine the size three sets of quartet topologies and to recursively branch on those sets which are not tree-consistent, as will be explained in Section 5.

Proposition 1 *(Proposition 2 in [2]) Given a set of taxa S and a complete set of quartet topologies Q_S over these taxa, Q_S is tree-like iff the following so-called*

substitution property *holds for every five distinct taxa* $a, b, c, d, e \in S$*:*

$$[ab|cd] \in Q_S \text{ implies } [ab|ce] \in Q_S \text{ or } [ae|cd] \in Q_S.$$

In the following, we show that in Proposition 1, we can replace the substitution property introduced by Bandelt and Dress with the more common term of tree-consistency. This is because, for an incomplete set of only three topologies, the substitution property is tightly connected to the tree-consistency of the topologies. We will state this in the following technical Lemmas 1 and 2 (proofs omitted, see [10]) and later use it to give, in Theorem 1, another interpretation of Proposition 1.

Lemma 1 *Three topologies involving more than five taxa are tree-consistent.*

When searching for local conflicts, Lemma 1 makes it possible to focus on the case of three topologies involving only five taxa. If the substitution property, as given in Proposition 1, is not satisfied, we say that the topologies for the quartets $\{a, b, c, d\}$, $\{a, b, c, e\}$, and $\{a, c, d, e\}$ *contradict* the substitution property.

Lemma 2 *For a given a set of taxa* S*, three topologies consisting of taxa from* S *are tree-consistent iff they do not contradict the substitution property.*

Note that Lemma 2 involving a necessarily incomplete set of three topologies does not generalize from size three to an incomplete set of arbitrary size, as exhibited in the following example. For taxa $\{a, b, c, d, e, f\}$, consider the incomplete set of topologies $[ab|cd]$, $[ab|ce]$, $[bc|de]$, $[cd|ef]$, and $[af|de]$. Without going into the details, we only state here that these topologies are not tree-consistent, although there are no three topologies which contradict the substitution property.

Theorem 1 now will make it clearer that "global" tree-consistency of a complete set of topologies reflects in "local" tree-consistency of every three topologies taken from this set.

Theorem 1 *Given a set of taxa* S *and a complete set of quartet topologies* Q_S *over* S*,* Q_S *is tree-like (and, thus, tree-consistent) iff every set of three topologies from* Q_S *is tree-consistent.*

Proof. Due to Lemma 2 we may replace the substitution property in Proposition 1 with tree constistency. This gives the result. □

When we have a complete set of topologies Q_S for a set of taxa S, we do not necessarily know whether the set is tree-like or not. If it is not, we can, according to Theorem 1, track down a subset of three topologies that is not tree-consistent. Our goal will be to detect all these local conflicts. This will be the preprocessing stage of the algorithm that will be described in Section 5, in order to (try to) "repair" the conflicts in a succeeding stage of the algorithm. We can find all these local conflicts in time $O(n^5)$ as follows. Since, following Lemma 1, only three topologies involving five taxa can form a local conflict, it suffices to consider all size five sets of taxa $\{a, b, c, d, e\} \subseteq S$. There are five quartets over this size five

set of taxa, namely, $\{a, b, c, d\}$, $\{a, b, c, e\}$, $\{a, b, d, e\}$, $\{a, c, d, e\}$, and $\{b, c, d, e\}$. For the topologies of these quartets, we can test, in constant time, whether there are three among them that are not tree-consistent. Doing so for every size five set, we will, if Q_S is not tree-consistent, certainly obtain a size three subset of Q_S which is not tree-consistent. Moreover, from Lemma 2 we know that we find *all* these local conflicts in time $O(n^5)$.

We can improve this time bound for the preprocessing stage of the algorithm to be described in Section 5 with the following result by Bandelt and Dress [2]. They show that it is sufficient to restrict our attention to the size five sets containing some arbitrarily fixed taxon f.

Proposition 2 *(Proposition 6 in [2]) Given a set of taxa S, a complete set of quartet topologies Q_S, and some taxon $f \in S$, then Q_S is tree-like iff every size five set of taxa which contains f satisfies the substitution property.*

Following Proposition 2, we can select some arbitrary $f \in S$ and examine only the size five sets involving f. Similar to our procedure described above, we consider every such size five set containing f separately. Among the topologies over this size five set, we search the size three sets which are not tree-consistent. If the set of quartet topologies Q_S is not tree-consistent, we will find a size three set of quartet topologies which is not tree-consistent. Finding these local conflicts which involve f can be done in time $O(n^4)$.

4 Combinatorial characterization of local conflicts

Given three topologies, we need to decide whether they are tree-consistent or not. Directly using the definition of tree-consistency turns out to be a rather technical, troublesome task, since we have to reason whether or not a tree topology exists that induces the topologies. Similarly, it can be difficult to test, for the topologies, whether or not they contradict the substitution property. To make things less technical and easier to grasp, we subsequently give a useful combinatorial characterization of local conflicts, i.e., three topologies which are not tree-consistent. Note that in the following definition, we distinguish two possible *orientations* of a quartet topology $[ab|cd]$, namely, $[ab|cd]$, with a, b on its left hand side and c, d on its right hand side, and $[cd|ab]$, with the sides interchanged.

Definition 1. *Given a set of topologies where each of the topologies is assigned an orientation, let l be the number of different taxa occurring in the left hand sides of the topologies and let r be the number of different taxa occurring in the right hand sides of the topologies. The* signature*, then, is the pair (l, r) that, over all possible orientations for these topologies, minimizes l.*

Theorem 2 *Three quartet topologies are not tree-consistent iff they involve five taxa and their signature is $(3, 4)$ or $(4, 4)$.*

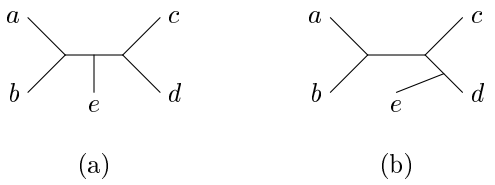


Fig. 1. Possible trees for $[ab|cd]$ and taxon e in the proof of Theorem 2.

Proof. ($\Rightarrow$) We show that, given three topologies t_1, t_2, and t_3 which are not tree-consistent, they involve five taxa and have signature $(3,4)$ or $(4,4)$. From Lemma 2, we know that three topologies are not tree-consistent iff they contradict the substitution property. To recall, three topologies contradict the substitution property if, for one of these topologies, w.l.o.g., $t_1 = [ab|cd]$, neither the topology t_2 for quartet $\{a,b,c,e\}$ is $[ab|ce]$ nor the topology t_3 for quartet $\{a,c,d,e\}$ is $[ae|cd]$. Therefore, the topology t_2 is either $[ac|be]$ or $[ae|bc]$, and the topology t_3 is either $[ac|de]$ or $[ad|ce]$. By exhaustively checking the possible combinations, we can find that the topologies involve five taxa and their signature is $(3,4)$ (e.g., for $t_2 = [ac|be]$ and $t_3 = [ac|de]$) or $(4,4)$ (e.g., for $t_2 = [ac|be]$ and $t_3 = [ad|ce]$).

($\Leftarrow$) We are given three topologies, t_1, t_2, and t_3, involving five taxa and having signature $(3,4)$ or $(4,4)$. Assume that they are tree-consistent. Showing that this implies signature $(2,3)$ or $(3,3)$, we prove that the assumption is wrong. For tree-consistent t_1, t_2, and t_3, we can find a tree inducing them. With, w.l.o.g., taxa $\{a,b,c,d,e\}$ and $t_1 = [ab|cd]$, we mainly have two possibilities: we can attach the leaf e on the middle edge of topology t_1, as shown in Figure 1(a), or we can attach e on one of the four side branches of t_1, as exemplarily shown in Figure 1(b). Considering the sets of quartet topologies induced by these trees, we find, in each case, that the set has signature $(3,3)$. For instance, the topologies induced by the tree in Figure 1(a) are, besides t_1, $[ab|ce]$, $[ab|de]$, $[ae|cd]$, and $[be|cd]$. Three topologies selected from these have signature $(3,3)$ (e.g., $[ab|cd]$, $[ab|ce]$, and $[ae|cd]$) or $(2,3)$ (e.g., $[ab|cd]$, $[ab|ce]$, and $[ab|de]$). □

Using Theorem 2, we can determine whether three topologies are conflicting by simply counting the involved taxa and computing their signature.

5 Fixed parameter algorithm for MQI

In this section, we present a recursive algorithm solving MQI with parameter k. Before calling the recursive part for the first time, one has to build the list of size three sets of quartets whose topologies are not tree-consistent. The preparation of this *conflict list* is explained in Section 3. After that, we call the recursive procedure of the algorithm with argument k.

The recursive procedure selects a local conflict to branch on from the conflict list. This branching is done by changing one topology from the selected local conflict, updating the conflict list, and calling the recursive procedure with argument $k-1$ on the thereby created subcases. We will later explain how to select and change the topologies when branching. After a topology t has changed, the algorithm updates the conflict list as follows: It (1) removes the size three sets of quartets in the list whose topologies are now tree-consistent, and (2) adds the size three sets of quartets not in the list whose topologies now form a local conflict.

The recursion stops if no conflicts are left in the conflict list (we have found a solution), or if $k = 0$ (in case the conflict list is not empty, we did not find a solution in this branch of the search tree). When a solution is found, the algorithm outputs the current set of topologies, i.e., a complete set of quartet topologies that is tree-like and that can be obtained by altering at most k topologies in the given set of topologies. From this tree-like set of quartet topologies, it is possible to derive the evolutionary tree in time $O(n^4)$ [4]. Thus scanning the whole search tree, we find *all* solutions that we can obtain by altering at most k topologies.

Running time. For establishing an upper bound on the running time, we consider the preprocessing, the update procedure, and the size of the search tree. The preprocessing can be done in time $O(n^4)$, as explained in Section 3.

Updating the conflict list can be done in time $O(n)$: Following Lemma 1, local conflicts can only occur among three topologies consisting of no more than five taxa. Therefore, having changed the topology of one quartet $\{a, b, c, d\}$, we only have to examine the "neighborhood" of the quartet, i.e., those sets of five taxa containing a, b, c, d. For every such set of five taxa, it can be examined in constant time whether for three topologies over the five taxa, a new conflict emerged, or whether an existing conflict has been resolved. Given taxa a, b, c, d, we have $n - 4$ choices for a fifth taxon. Thus, $O(n)$ is an upper bound for the update procedure.[3]

Now, we consider the search tree size. By a careful selection of subcases to branch into, we can find a way to make at most four recursive calls on an arbitrarily selected local conflict, i.e., for every three topologies which are not tree-consistent. Let t_1, t_2, and t_3 be three topologies which are not tree-consistent, and let, w.l.o.g., $t_1 = [ab|cd]$. Following Lemma 1, the topologies involve only one additional taxon, say e. Following Lemma 2, t_1, t_2, t_3 contradict the substitution property. Given $t_1 = [ab|cd]$, the substitution property requires topology $[ab|ce]$ or topology $[ae|cd]$. Therefore, we can, w.l.o.g., assume the following setting for three quartets contradicting the substitution property: Topology $t_1 = [ab|cd]$, topology t_2 is the topology for quartet $\{a, b, c, e\}$ different from $[ab|ce]$, and topology t_3 is a topology for quartet $\{a, c, d, e\}$ different from $[ae|cd]$. In order

[3] In fact, as explained in Section 3, we only consider sets of five species containing a designated taxon f. Therefore, if we change the topology of a quartet $\{a, b, c, d\}$ which does not contain the designated taxon f, then we only have to consider *one* set of five topologies, namely, $\{a, b, c, d, f\}$. In this special case, the update procedure can be done in time $O(1)$.

to change the three topologies to satisfy the substitution property, we have the following possibilities. We can change t_1; either (1) we change t_1 to $[ac|bd]$, or (2) we change t_1 to $[ad|bc]$. Otherwise, we can assume that t_1 is not changed. Then, we have to (3) change t_2 to $[ab|ce]$ or (4) change t_3 to $[ae|cd]$, because these are the only remaining possibilities to satisfy the substitution property. Since the height of the search tree is at most k, the preceding considerations justify an upper bound of 4^k on the exponential growth and yield the following theorem, which summarizes our findings.

Theorem 3 *The* MQI *problem can be solved in time* $O(4^k \cdot n + n^4)$.

Note that this running time is not only true for the algorithm reporting *one* solution, but also for reporting *all* evolutionary trees satisfying the requirement. Our algorithm has $O(kn^4)$ memory requirement, where the input size is already $O(n^4)$. The correctness of the algorithm follows easily from Theorem 1.

6 Improving the running time in practice

Besides improving the worst case bounds on the algorithm's running time, we can also extend the algorithm in order to improve the running time in practice without affecting the upper bounds. In this section, we collect some ideas for such heuristic improvements.

Fixing topologies. It does not make sense to change a topology which, at some previous level of recursion, has been altered, or for which we explicitly decided not to alter it. If we decide not to alter a topology in a later stage of recursion, we call this *fixing* the topology. This avoids redundant branchings in the search tree.

Forcing topologies to change. It might be possible to identify topologies which necessarily have to be altered in order to find a solution. We call this *forcing* a topology to change. The ideas described here are similar to those used in the so-called reduction to problem kernel for the 3-Hitting Set problem [13].

Lemma 3 *Consider an instance of the* MQI *problem in which quartet* q *has topology* t*. If there are more than* $3k$ *distinct local conflicts which contain* t *then, in a solution for this instance, the topology for* q *is different from* t*.*

Proof. In Section 3, we showed that three topologies only can form a local conflict if there are not more than five taxa occurring in them (see Lemma 1). For five taxa, there are five quartets consisting of these taxa, e.g., for taxa $\{a, b, c, d, e\}$ the quartets are $\{a, b, c, d\}$, $\{a, b, c, e\}$, $\{a, b, d, e\}$, $\{a, c, d, e\}$, and $\{b, c, d, e\}$. Therefore, when given two quartet topologies t_1 and t_2, we make the following observations. If there are more than five taxa occurring in t_1 and t_2, they cannot form a conflict with a third topology. If there are exactly five taxa occurring in t_1 and t_2, then there are five quartets consisting of these five taxa, two of which are the quartets for t_1 and t_2. The remaining three topologies are the only possibilities for a topology t_3 that could form a conflict with t_1 and t_2.

Now, consider the situation in which, for a quartet topology t, we have more than $3k$ distinct local conflicts which contain t. From the preceding discussion, we know that for any t', there are at most three topologies such that t and t' can form a conflict with it. Consequently, there must be more than k distinct topologies t' that occur in a local conflict with t. We show by contradiction that we have to alter topology t to find a solution. Assume that we can find a solution while not altering t. By changing a topology t', we can cover at most three conflicts, since there are at most three local conflicts containing both t and t'. Therefore, by changing k topologies, we can resolve at most $3k$ local conflicts. This contradicts our assumption and shows that we have to alter t to find a solution. □

Recognizing hopeless situations. Now, we describe situations in which, at some level in the search tree when we are allowed to alter at most k topologies, we can recognize that we cannot find a solution. Thus, we can "cut off," i.e., omit, complete subtrees of the search tree.

Having a local conflict consisting only of fixed topologies, we obviously cannot resolve this conflict while not changing one of the fixed topologies. As another observation, we know that for a solution, we have to change the forced topologies. If after identifying these forced topologies, there are more than k of them, it is obvious that a solution is not possible—already by changing these topologies, we would change more topologies than we are allowed to.

The following two lemmas contain more involved observations. Their proofs use similar ideas as used in the proof of Lemma 3 (see [10]). If a local conflict does not contain a topology which is forced to change, then we call it an *unforced* local conflict.

Lemma 4 *Let us have an instance of the* MQI *problem in which we have identified p conflicts which are forced to change. If the number of unforced local conflicts is greater than $3(k-p)k$, then the instance has no solution.*

Lemma 5 *An instance of the* MQI *problem in which the number of local conflicts is greater than $6(n-4)k$ has no solution.*

Clever branching. Applying the rules described above will also significantly improve our situation when branching. For the general branching situation on a local conflict, we have shown in Section 5 that it is sufficient to branch into four subcases. Regarding topologies forced to change, we can, however, reduce the number of subcases. When we have identified a topology t which is forced to change, it is sufficient to branch into two subcases: one for each alternative topology of t. Regarding fixed topologies, we can take advantage of local conflicts which contain fixed topologies. Having a local conflict with one or two fixed topologies, we omit the subcases which change a fixed topology. This will reduce the number of subcases to three, two, or even one subcase.

Preprocessing by the Q*-method. The algorithmic improvements described above do not sacrifice the guarantee to find the optimal solutions. Using these improvements, we will find every solution that we would find without them. This

is not true for the following idea. We propose to use the Q*-method described by Berry and Gascuel [4] as a preprocessing for our algorithm. The Q*-method produces the maximum subset of the given quartet topologies that is tree-like. In the combined use with our algorithm, we fix these quartet topologies from the beginning. Therefore, our algorithm will compute the minimum number of quartet topologies we have to change in order to obtain a tree-like set of topologies that contains the topologies fixed by the Q*-method. The tree we obtain will be a refinement of the tree reported by the Q*-method which may contain unresolved branches. Thus, we cannot guarantee that the reported tree is the optimal solution for the MQI problem. On real data, however, it is the optimal tree with high certainty: Suppose it is not. Then there are four taxa a,b,c, and d that are arranged in another way by the Q*-method than they would be arranged in the optimal solution for the MQI problem. As we are working on a complete set of topologies, this would imply that there are at least $n-3$ quartets that would make the same wrong prediction for the arrangement of a, b, c, d: the quartet $\{a, b, c, d\}$ and, for all $e \in S - \{a, b, c, d\}$, one quartet over $\{a, b, c, d, e\}$ that involves e. On real data, this is very unlikely. Our experiments described in Section 8 support the conjecture that with the preprocessing by the Q*-method, we find every solution that the MQI algorithm would find. Moreover, the experiments show that this enhancement allows us to process much larger instances than we could without using it.

7 Related problems

We now come to some variants and generalizations of the basic MQI problem and their fixed parameter tractability. These variations arise in practice due to the fact that often quartet inference methods cannot non-ambiguously predict a topology for every quartet. Perhaps the most natural generalization of MQI is to consider weighted quartet topologies.

Weighted MQI. Weights arise since a quartet inference method can predict the topology for a quartet with more or less certainty. Therefore, we can assign weights to the quartet topologies reflecting the certainty they are predicted with. Given a complete set of weighted topologies Q_S and a positive integer k, we distinguish two different questions.

1. Assume that we are given a complete set of weighted topologies Q_S, with positive real weights, and a positive integer k. A binary tree is a candidate for a solution if the set of quartet topologies induced by this tree differs from Q_S in the topologies for at most k quartets. Can we, among all candidate trees satisfying this property, find the one such that the topologies in Q_S which are not induced by the tree have minimum total weight?
   The algorithm in Section 5 can compute *all* solution trees. So, we can, without sacrificing the given time bounds, find this tree among the solution trees for which the "wrong" quartet topologies have minimal total weight.
2. Assume that we are given a complete set of weighted topologies Q_S, each topology having a real weight ≥ 1, and a positive real K. Is there a binary

tree such that the quartet topologies induced by the tree differ from the given topologies only for topologies having total weight less than K?
Again, we can use the algorithm presented in Section 5. When branching into different subcases, the time analysis of the algorithm relied on the fact that in each subcase at least one quartet topology is changed, i.e., added to the "wrong" topologies. In the current situation of weighted topologies with weights ≥ 1, each subcase changes quartet topologies having a total weight of at least 1. The time analysis of our algorithm is, therefore, still valid and the time bounds remain the same.

Allowing arbitrarily small weights in question 2, the problem cannot be fixed parameter tractable, unless P = NP. To see this, take an instance of unweighted MQI with parameter k. We can turn this instance into an instance of weighted MQI by assigning all topologies weight $1/k$ and setting the parameter to 1. A fixed parameter algorithm for the problem with arbitrary weights > 0 would thus give a polynomial time solution for MQI, which contradicts the NP-completeness of MQI unless P = NP. Having, however, weights of size at least ϵ for some positive real ϵ, the problem is fixed parameter tractable as we described here for the special case that $\epsilon = 1$ (similar to Weighted Vertex Cover in [14]).
Underspecified MQI. Due to lack of information or due to ambiguous results, a quartet inference method may not be able to compute a topology for every quartet, so there may be quartets for which no topology is given. Assuming a bounded number of quartets with missing topology, we formulate the problem as follows. Given a set S of taxa, integers k and k', and a set of topologies Q_S, such that Q_S contains quartet topologies for all quartets over S except for k' many. Then, we ask whether there is a binary tree such that the quartet topologies induced by the tree differ from the given topologies only for k topologies.

The set of topologies is "underspecified" by k' topologies. We can solve the problem as follows. Having three possible topologies for each quartet, we can, for a quartet without given topology, branch into three subcases, one for each of its three possible topologies. Having selected a topology for each such quartet, we run the algorithm from Section 5. The resulting algorithm has time complexity $O(3^{k'} \cdot 4^k \cdot n + n^5)$ and shows that the problem is fixed parameter tractable for parameters k and k'. Note that for unbounded k' this problem is NP-complete even for $k = 0$ [16] and, therefore, is not fixed parameter tractable.

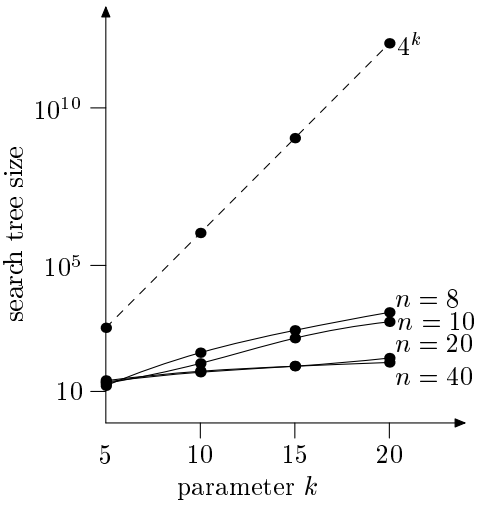
We only briefly mention another variant of MQI, *Overspecified* MQI: In that problem, we are, compared to MQI, given an additional integer k'' and two topologies instead of one for k'' many quartets. For these quartets, we are free to choose one of the given topologies. In a similar way as for underspecified MQI, we can show that overspecified MQI is fixed parameter tractable for parameters k and k''.

8 Experimental evaluation

To investigate the usefulness and practical relevance of the algorithm for unweighted MQI, we performed experiments on artificial as well as on real data

n	k	time in sec	size of search tree
10	5	0.03	16
	10	0.05	93
	20	0.61	1792
20	5	0.19	20
	10	0.20	41
	20	0.25	115
	40	16.83	20375
50	5	9.21	20
	10	9.01	46
	20	8.63	92
	40	23.25	178
	100	2409.07	653571

(a)



(b)

Fig. 2. Comparing running time and search tree size for different values of n and k.

from fungi. The implementation of the algorithm was done using the programming language C. The algorithm contains the enhancements described in Section 6. The combined use with the Q*-method was, however, only applied when processing the fungi data, not when processing the artificial data. The reported tests were done on a LINUX PC with a Pentium III 750 MHz processor and 192 MB main memory.

8.1 Artificial data

We performed experiments on artificially generated data in order to find out which kind of data sets our algorithm can be especially useful for. For a given number n of taxa and parameter k, we produce a data file as follows. We generate a random evolutionary tree for n taxa and derive the quartet topologies from that tree. Then, we change k distinct, arbitrarily selected topologies in a randomly chosen way. This results in an MQI instance that certainly can be solved with parameter k. For each pair of values for n and k, ten different data sets were created. The reported results are the average for test runs on ten data sets.

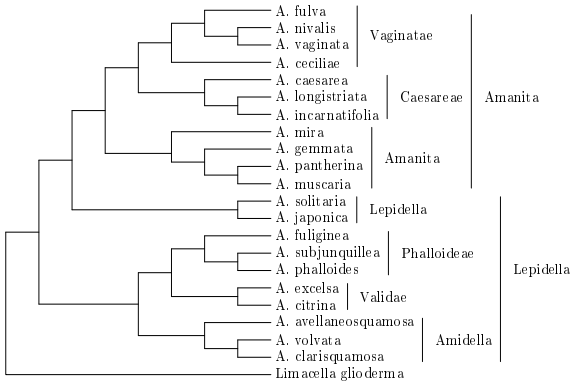
We experimented with different values of n and k. As a measure of performance, we use two values: We report the processing time and, since processing time is heavily influenced by system conditions, e.g., memory access time in case of cache faults, also the search tree size. The search tree size is the number of the search trees nodes, both inner nodes and leaves, and it reflects the exponential growth of the algorithm's running time.

Figure 2(a) gives a table of results for different values of n and k. Regarding the processing time, we note, on the one hand, the increasing time for fixed n and

n	running time in sec. no Q*	with Q*
8	0.46	0.36 (21%) [a]
9	3.41	0.85 (32%)
10	35.96	2.68 (38%)
11	617.56	4.11 (41%)
12	7039.82	5.44 (43%)

[a] Percentage of quartet topologies fixed by Q*

(a)



(b)

Fig. 3. (a) Speed-up when using Q* preprocessing. (b) Optimal tree found for a set of 21 *Amanita* species and one outgroup taxon; indicated is the grouping of *Amanita* species into 7 sections and 2 subgenera.

growing k. On the other hand, we observe that for moderate values of k, we can process large instances of the problem, e.g., $n = 50$ and $k = 100$ in 40 minutes. For comparison of the algorithm's performance, consider the results reported by Ben-Dor *et al.* [3], who solve MQI instances also giving guaranteed optimal results. They only report about processing up to 20 taxa and list, admittedly for a high number of erroneous topologies, a running time of 128 hours for this case (on a SUN Ultra-4 with 300 MHz).

In Figure 2(b) we compare, on a logarithmic scale, the theoretical upper bound of 4^k to the real size of the search tree. For each fixed number of taxa n, we give a graph displaying the growth of search tree size for increasing k. The search trees are, by far, smaller than the 4^k bound. This is mainly due to the practical improvements of the algorithm (see Section 6). We also note that for equal value of k, a higher number n of taxa often results in a smaller search tree.

8.2 Real data

Using our algorithm, we analyzed the evolutionary relationships of species from the mushroom genus *Amanita*, a group that includes well-known species like the Fly Agaric and the Death Cap. The underlying data are an alignment of nuclear DNA sequences coding for the D1/D2 region of the ribosomal large subunit (alignment length 576) from *Amanita* species and one outgroup taxon, as used by Weiß *et al.* [18]. We inferred the quartet topologies by (1) using `dnadist` from the Phylip package [9] to compute pairwise distances with the maximum likelihood metric, and (2) using `distquart` from the Phyloquart package [4] to infer quartet topologies based on the distances.

The analysis was done by a preprocessing of the data using the Q*-method, also taken from the Phyloquart package. Experiments on small instances, e.g.,

10 taxa, show that all solutions we find without using the Q*-method are also found when using it. Using the Q*-method, however, results in a significant speed-up of the processing. Figure 3(a) shows this impact for small numbers of *Amanita* species. Note, however, that the speed-up heavily depends on the data. In Figure 3(a) and in the following, we neglect the time needed for the preprocessing by the Q*-method, which is, e.g., 0.11 seconds for $n = 12$.

We processed a set of $n = 22$ taxa in 35 minutes. The resulting tree was rooted using the outgroup taxon *Limacella glioderma* and is displayed in Figure 3(b). We found the best solution for $k = 979$ for the given 7315 quartet topologies. The Q*-method had fixed 41 percent of the quartet topologies in advance. Considering the tree, the grouping of taxa is consistent with the grouping into seven sections supported by Weiß *et al.* [18], who used the distance method neighbor joining, heuristic parsimony methods, and maximum likelihood estimations. Particularily, our grouping is nearly identical to the topology revealed by Weiß *et al.* using maximum likelihood estimation. This topology is well compatible with classification concepts based on morphological characters, e.g., the sister group relationship of sections *Vaginatae* and *Caesareae*, and the monophyly of subgenus *Amanita*.

One might hope that quality of quartet inference techniques will improve in the future. This would lead to instances requiring smaller values of k.

9 Conclusion

We showed that the Minimum Quartet Inconsistency problem can be solved in worst case time $O(4^k n + n^4)$ when parameter k is the number of faulty quartet topologies. This means that the problem is fixed parameter tractable. Several ideas for tuning the algorithm show that the practical performance of the algorithm is much better that the theoretical bound given above. This is clearly expressed by our experimental results. Note that there is an ongoing discussion about the usefulness of quartet methods: St. John *et al.* [15] give a rather critical exposition of the practical performance of quartet methods (in particular, quartet puzzling) in comparison with the neighbor joining method, which is in opposition to results reported by Strimmer and v. Haeseler [17].

Concerning future work, we want to extend our experiments to weighted quartet topologies and to other data. Also, the fact that we can obtain all optimal and near-optimal solutions and the usefulness of this deserves further investigation. From a parameterized complexity point of view, it remains an open question to find a so-called reduction to problem kernel (see [1, 7, 8] for details). The further reduction of the tree size concerning theoretical, as well as experimental bounds, is a worthwhile future challenge.

Acknowledgment. We are grateful to Michael Weiß from the Biology Department (Systematic Botany and Mycology group), Universität Tübingen, for providing us with the *Amanita* data, supporting us in the interpretation of our results, and many constructive remarks improving the presentation significantly.

References

1. J. Alber, J. Gramm, and R. Niedermeier. Faster exact solutions for hard problems: a parameterized point of view. *Discrete Mathematics*, 229(1-3):3–27, 2001.
2. H.-J. Bandelt and A. Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7:309–343, 1986.
3. A. Ben-Dor, B. Chor, D. Graur, R. Ophir, and D. Pelleg. Constructing phylogenies from quartets: elucidation of eutherian superordinal relationships. *Journal of Computational Biology*, 5:377–390, 1998.
4. V. Berry and O. Gascuel. Inferring evolutionary trees with strong combinatorial evidence. *Theoretical Computer Science*, 240:271–298, 2000. Software available through `http://www.lirmm.fr/~vberry/PHYLOQUART/phyloquart.html`.
5. P. Buneman. The recovery of trees from measures of dissimilarity. In Hodson *et al.*, eds, *Anglo-Romanian Conference on Mathematics in the Archaeological and Historical Sciences,* pages 387–395, 1971. Edinburgh University Press.
6. B. Chor. From quartets to phylogenetic trees. In *Proceedings of the 25th SOFSEM*, number 1521 in LNCS, pages 36–53, 1998. Springer.
7. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* 1999. Springer.
8. M. R. Fellows. Parameterized complexity: new developments and research frontiers. In Downey and Hirschfeldt, eds, *Aspects of Complexity,* to appear, 2001. De Gruyter.
9. J. Felsenstein. PHYLIP (Phylogeny Inference Package) version 3.5c. Distributed by the author. Department of Genetics, University of Washington, Seattle. 1993. Available through `http://evolution.genetics.washington.edu/phylip`.
10. J. Gramm and R. Niedermeier. Minimum Quartet Inconsistency is fixed parameter tractable. Technical Report WSI-2001-3, WSI für Informatik, Universität Tübingen, Fed. Rep. of Germany, January 2001. Report available through `http://www-fs.informatik.uni-tuebingen.de/~gramm/publications`.
11. T. Jiang, P. Kearney, and M. Li. Some open problems in computational molecular biology. *Journal of Algorithms*, 34:194–201, 2000.
12. T. Jiang, P. Kearney, and M. Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. To appear in *SIAM Journal on Computing*, 2001.
13. R. Niedermeier and P. Rossmanith. An efficient fixed parameter algorithm for 3-Hitting Set. Technical Report WSI-99-18, WSI für Informatik, Universität Tübingen, October 1999. To appear in *Journal of Discrete Algorithms*.
14. R. Niedermeier and P. Rossmanith. On efficient fixed parameter algorithms for Weighted Vertex Cover. In *Proceedings of the 11th International Symposium on Algorithms and Computation*, number 1969 in LNCS, pages 180–191, 2000. Springer.
15. K. St. John, T. Warnow, B.M.E. Moret, and L. Vawter. Performance study of phylogenetic methods: (unweighted) quartet methods and neighbor-joining. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 196–205, 2001. SIAM Press.
16. M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.
17. 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.
18. M. Weiß, Z. Yang, and F. Oberwinkler. Molecular phylogenetic studies in the genus Amanita. *Canadian Journal of Botany*, 76:1170–1179, 1998.