Kernel and Fast Algorithm for Dense Triplet Inconsistency

Sylvain Guillemot and Matthias Mnich

1 Lehrstuhl für Bioinformatik, Friedrich-Schiller Universität Jena, Ernst-Abbe Platz 2, 00743 Jena, Germany
sylvain.guillemot@uni-jena.de
2 Department of Mathematics and Computer Science, Technische Universität Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
m.mnich@tue.nl

Abstract. We study the parameterized complexity of inferring supertrees from sets of rooted triplets, an important problem in phylogenetics. For a set $L$ of labels and a dense set $R$ of triplets distinctly leaf-labeled by 3-subsets of $L$ we seek a tree distinctly leaf-labeled by $L$ and containing all but at most $p$ triplets from $R$ as homeomorphic subtree. Our results are the first polynomial kernel for this problem, with $O(p^2)$ labels, and a subexponential fixed-parameter algorithm running in time $2^{O(p^{1/3} \log p)} + O(n^4)$.

1 Introduction

In phylogenetics, distinctly leaf-labeled trees represent the evolutionary history of a set of species, each species corresponding to a label. Supertree methods are widely used in this field, in order to construct a large tree from smaller trees on overlapping subsets of species. The simplest approach in this setting consists in inferring the smallest possible informative trees. Such trees are either rooted triplets, rooted binary trees on three labels, or quartets, unrooted ternary trees on four labels. Quartet methods received prominent attention over the last decade, whereas triplets methods were somewhat overlooked though they enjoy similar interesting properties and may be less computationally expensive.

Let $L$ be a set of labels. A set $R$ of rooted binary (unrooted ternary) trees distinctly leaf-labeled by subsets of $L$ is consistent if there exists a rooted binary (unrooted ternary) tree distinctly leaf-labeled by $L$ and containing every element of $R$ as homeomorphic subtree; and inconsistent otherwise. Deciding consistency for a set of triplets is polynomial-time solvable [1]; in contrast, this problem is NP-hard for quartets [19]. For an inconsistent triplet set $R$, two approaches have been studied to obtain a consistent triplet set from it.

The first approach is to find a minimum-cardinality subset $L' \subseteq L$ such that removing all leaves labeled by elements from $L'$ from triplets in $R$ yields a
consistent set of trees. The problem of finding \( L' \) is the dual of the maximum agreement supertree problem \([\text{16}]\text{[14]}\), and we refer to it as Minimum Label Inconsistency (MLI). The parameterization of MLI by \(|L'|\) is denoted as p-MLI; this problem is \( \text{NP} \)-hard and fixed-parameter intractable \([\text{6}]\). The restriction of p-MLI to instances that are dense, that is \( \mathcal{R} \) contains exactly one triplet for each 3-subset of \( L \), is fixed-parameter tractable; call this problem p-DENSE MLI. Fixed-parameter tractability of p-DENSE MLI follows from that for dense triplet sets, consistency has a characterization in terms of obstructions involving at most four labels and three triplets. More precisely, for a dense triplet set \( \mathcal{R} \) on \( n \) labels, we can build in time \( O(n^4) \) the set of obstructions, and in time \( O(n^3) \) either find an obstruction or decide if \( \mathcal{R} \) is consistent \([\text{14}]\). These results lead to a \( O(4^p n^3) \)-time algorithm for p-DENSE MLI \([\text{14}]\).

A fixed-parameter algorithm. The other approach is to remove a minimum-size set \( \mathcal{R}' \) of triplets from \( \mathcal{R} \) such that the set \( \mathcal{R} \setminus \mathcal{R}' \) is consistent. The problem of finding \( \mathcal{R}' \) is the Rooted Triplet Inconsistency (RTI) problem \([\text{9}]\text{[10]}\). We denote the restriction of RTI to dense instances, and parameterized by the size of \( \mathcal{R}' \), by p-DENSE RTI. This restriction of RTI is still \( \text{NP} \)-hard \([\text{10}]\). Our first result is a simple \( O(4^p n^3) \)-time algorithm for p-DENSE RTI, based on characterization of consistency in terms of obstructions and given in Section 3. For general instances, the parameterization of RTI by \(|\mathcal{R}'|\) is not fixed-parameter tractable unless some unlikely collapse of complexity classes occurs \([\text{10}]\).

A polynomial kernel. Any fixed-parameter tractable problem \( \Pi \) has a kernelization algorithm, or kernel, which is an algorithm that given a pair \( (x, p) \) outputs in time polynomial in \(|x| + p\) a pair \( (x', p') \) such that \( (x, p) \in \Pi \) if and only if \( (x', p') \in \Pi \) and \(|x'|, p' \leq g(p)\), where \( g \) is some computable function. The function \( g \) is referred to as the size of the kernel, and if \( g(p) = p^{O(1)} \) then \( \Pi \) is said to admit a polynomial kernel. Kernels are important to practically solve instances of \( \text{NP} \)-hard problems through data reduction, but not every fixed-parameter tractable problem admits a polynomial kernel \([\text{8}]\). Our second result is the first polynomial kernel for p-DENSE RTI: in Section 4 we describe a kernelization algorithm which produces in time \( O(n^4) \) a kernel with \( O(p^2) \) labels.

Subexponential fixed-parameter algorithm. While interesting by itself, our polynomial kernel serves as the basis for our third result. In Section 5 we present a subexponential fixed-parameter algorithm for p-DENSE RTI. The algorithm applies the method of chromatic coding, which was recently introduced by Alon et al. \([\text{3}]\) to solve the tournament feedback arc set problem (FAST) in time \( O(n^3 + 2^{O(p^{1/2} \log p)}) \). Similarly, chromatic coding provided a subexponential fixed-parameter algorithm for the DENSE BETWEENNESS problem parameterized by solution size \( p \), running in time \( O(n^4 + 2^{O(p^{1/3} \log p)}) \) \([\text{18}]\). Let us remark that the trees we are constructing possess a more complex structure than the linear orderings encountered in ranking problems such as FAST and BETWEENNESS.

Chromatic coding is a variant of the color coding technique by Alon et al. \([\text{4}]\). It requires several ingredients: (i) a kernel of quadratic size, (ii) an algorithm solving instances of size \( n \) and colored with \( k \) colors in time \( n^{O(k)} \), (iii) a coloring