Discrete Applied Mathematics 160 (2012) 1634–1638

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

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

Note

A note on the parameterized complexity of unordered maximum tree orientation

Sebastian Böcker [a], Peter Damaschke [b,*]

[a] *Lehrstuhl für Bioinformatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany*

[b] *Department of Computer Science and Engineering, Chalmers University, S-41296 Göteborg, Sweden*

ARTICLE INFO

Article history:
Received 13 August 2011
Received in revised form 10 February 2012
Accepted 16 February 2012
Available online 9 March 2012

Keywords:
Graph orientation
Directed path
Parameterized reduction
Bipartization

ABSTRACT

In the Unordered Maximum Tree Orientation problem, a set P of paths in a tree and a parameter k is given, and we want to orient the edges in the tree such that all but at most k paths in P become directed paths. This is a more difficult variant of a well-studied problem in computational biology where the directions of paths in P are already given. We show that the parameterized complexity of the unordered version is between Edge Bipartization and Vertex Bipartization, and we give a characterization of orientable path sets in trees by forbidden substructures, which are cycles of a certain kind.

1. Introduction

Consider an undirected graph. To orient an edge uv means to decide on one direction, from u to v or vice versa. Edges can be oriented independently. Given an orientation of all edges, a path in the graph is called directed if we can traverse this path in one direction, following the orientations of its edges. For convenience we will use the terms *node* and *vertex* in a tree and in a general graph, respectively. Define the following problem:

Maximum Graph Orientation: Given is an undirected graph, a set P of n ordered pairs of vertices, and a parameter $k < n$. Delete at most k pairs in P and orient the edges in the graph such that, for each remaining pair $(u, v) \in P$, some directed path goes from u to v.

P can also be a multiset, that is, a pair (u, v) may appear several times in P. Then we do not demand several distinct paths from u to v, but the multiplicity of (u, v) serves as a weight when it comes to deletions. For notational convenience we will speak of a set P, although all considerations carry over to multisets.

A motivation of the problem is the inference of signal transmissions in protein–protein interaction networks, based on cause–effect pairs. For example, if an experiment that changes the amount of a protein u causes also a change at v, some information must flow from u to v. Since a cycle can always be oriented, we can successively shrink cycles to new vertices until a tree remains, thereby preserving the solution space. Thus it suffices to consider the problem for trees, see [8].

Maximum Tree Orientation: Given is a tree, a set P of n ordered pairs of nodes of the tree, and a parameter $k < n$. Delete at most k pairs in P and orient the edges in the tree such that, for each remaining pair $(u, v) \in P$, some directed path goes from u to v.

Maximum Tree Orientation is NP-hard [8], and several approximation results have also been derived there. Another standard direction of research on hard problems is the computation of optimal solutions by mildly exponential or

* Corresponding author. Tel.: +46 31 772 5405; fax: +46 31 772 3663.
E-mail addresses: sebastian.boecker@uni-jena.de (S. Böcker), ptr@chalmers.se (P. Damaschke).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.02.017

parameterized algorithms. We state a few basic notions and refer to [4,9] for general introductions. In exponential time bounds we use the O^* notation that omits polynomial factors. A problem is fixed-parameter tractable (FPT) if it can be solved in $p(n) \cdot f(k) = O^*(f(k))$ time where p is a polynomial and f any computable function.

Maximum Tree Orientation can be reduced to the classical FPT problem Vertex Cover [3]: Construct a graph $G = (P, E)$ with P as vertex set, and with edges between any two vertices whose corresponding tree paths in P have conflicting orientations; equivalence is obvious. Hence Maximum Tree Orientation is in FPT, and it can be solved in "Vertex Cover time" which is much faster than $O^*(2^k)$, see [2] for the latest bound.

In the present paper we study a more difficult problem:

Unordered Maximum Graph Orientation: Given is an undirected graph, a set P of n unordered pairs of vertices, and a parameter $k < n$. Delete at most k pairs in P and orient the edges in the graph such that, for each remaining pair $(u, v) \in P$, some directed path goes from u to v, or in the opposite direction.

Here we have to decide not only on the deleted pairs but also on the orientations of pairs that are kept. This unordered version comes into mind naturally, and it has a plausible motivation as well: In the protein–protein interaction network application, the data may be expression profiles, such that one can observe that certain pairs are in correlation, but one cannot see what is the cause and the effect.

Still the unordered problem inherits some basic properties of the ordered counterpart. Exactly as in the ordered case, we can always orient cycles and successively shrink them to new vertices, thereby preserving the solution space. Therefore, the actual problem to consider is the following.

Unordered Maximum Tree Orientation: Given is a tree T, a set P of n unordered pairs of nodes of T, and a parameter $k < n$. Delete at most k pairs in P and orient the edges in T such that, for each remaining pair $(u, v) \in P$, the (u, v)-path in T is a directed path from u to v, or in the opposite direction.

Since any pair of nodes in a tree is connected by a unique path, we will henceforth consider P as a set of paths, and the problem is to orient all but k paths in P by edge orientations.

Trivially, Unordered Maximum Tree Orientation can be solved in $O^*(3^n)$ time: Decide for each path in P one orientation or decide not to orient the path, and then check consistency. A little more thinking yields.

Proposition 1. Unordered Maximum Tree Orientation *is solvable in* $O^*(2^n)$ *time.*

Proof. We process the paths one-by-one in a certain order and decide for each path whether to orient it or not. It always suffices to consider one orientation: For the first path, both orientations lead to symmetric solutions, hence we can take either one. In the following steps, we always take a path from P that intersects paths that are already oriented, hence only one possible orientation is consistent. If no such path exists, then we start a new connected component, and again the first orientation is arbitrary. □

Essentially the same argument shows.

Proposition 2. Unordered Maximum Tree Orientation *with* $k = 0$ *is solvable in polynomial time.* □

Triviality for $k = 0$ provokes the question whether Unordered Maximum Tree Orientation is also fixed-parameter tractable in parameter k, like the ordered version. In this paper we give an affirmative answer. More specifically, we show that its complexity is between two established FPT graph problems:

Edge Bipartization: Given a graph and a parameter k, delete at most k edges such that the graph becomes bipartite.

Vertex Bipartization: Given a graph and a parameter k, delete at most k vertices such that the graph becomes bipartite.

The following statements are well known to be equivalent:

- G is a bipartite graph.
- The vertices of G can be colored white and black such that any two adjacent vertices have distinct colors.
- G does not contain odd cycles.

Due to the last property, solutions to Vertex Bipartization are also known as *odd cycle transversals*.

We will give simple parameterized reductions from Edge Bipartization to Unordered Maximum Tree Orientation, and from the latter problem to Vertex Bipartization. Since Vertex Bipartization is solvable in $O^*(3^k)$ time [11,7], so is Unordered Maximum Tree Orientation. The best known FPT algorithm for Edge Bipartization, based on iterative compression, runs in $O^*(2^k)$ time [6]. Besides these implicit complexity bounds for Unordered Maximum Tree Orientation, we also give structural characterizations of path sets in trees that can be oriented.

2. Unordered maximum tree orientation versus bipartization problems

We call a set P of paths in a tree *orientable* if Unordered Maximum Tree Orientation has a solution with $k = 0$. Proposition 2 says that this property can be checked in polynomial time. Now we give a characterization of orientable path sets in terms of an auxiliary graph. A similar construction was used in [1] for a graph drawing problem. Interestingly, the