Exact Bipartite Crossing Minimization under Tree Constraints ⋆

Frank Baumann [a] Christoph Buchheim [a] Frauke Liers [b]

[a] *Technische Universität Dortmund, Fakultät für Mathematik, Vogelpothsweg 87, 44227 Dortmund, Germany*

[b] *Universität zu Köln, Institut für Informatik, Pohligstraße 1, 50969 Köln, Germany*

Key words: tanglegram, crossing minimization, quadratic programming

---

1 Abstract

A *tanglegram* consists of a pair of (not necessarily binary) trees. Additional edges, called *tangles*, may connect the leaves of the first with those of the second tree. The task is to draw a tanglegram with a minimum number of tangle crossings while making sure that the trees are drawn crossing-free. This problem has relevant applications in computational biology, e.g., for the comparison of phylogenetic trees. Most existing approaches are only applicable for binary trees. In this work, we show that the problem can be formulated as a quadratic linear ordering problem (QLO) with side constraints. In [1] it was shown that, appropriately reformulated, the QLO polytope is a face of some cut polytope. It turns out that the additional side constraints do not destroy this property. Therefore, any polyhedral approach to max-cut can be used in our context. We show that our approach is very efficient in practice for both random and real-world binary as well as general tanglegrams.

2 Introduction

The task of drawing tanglegrams arises in several relevant applications, e.g., in computational biology for the comparison of phylogenetic trees. Moreover,

⋆ Financial support from the German Science Foundation (DFG) is acknowledged under contracts Bu 2313/1–1 and Li 1675/1–1.

tanglegrams occur when analyzing software projects in which a tree represents package, class and method hierarchies. This application yields tanglegrams on trees that are not binary in general [5]. Most of the literature is concerned with the case of binary trees and leaves that are in one-to-one correspondence. [4] showed the NP-hardness of tanglegram layout, even in the case of binary trees. Our approach, to be explained below, generalizes the exact integer-programming (IP) approach of [5] for binary tanglegrams. In the latter, minimizing the number of tangle crossings reduces to solving an unconstrained quadratic binary optimization problem, which is well-known to be equivalent to a maximum cut problem in some associated graph with an additional node [2]. In an undirected graph $G = (V, E)$, the cut $\delta(W)$ induced by a set $W \subseteq V$ is defined as the set of edges (u, v) such that $u \in W$ and $v \notin W$. If edge weights are given, the weight of a cut is the total weight of edges in the cut. Now the maximum cut problem asks for a cut of maximal weight or cardinality. While in the recent paper by [7] the focus is on binary instances, a fixed-parameter algorithm for general tanglegram instances is presented. According to our knowledge, this is the only algorithm that could deal with non-binary trees; however, no implementation or running times are provided making it impossible to evaluate its practical performance.

3 An Exact Model for General Tanglegrams

Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. The task is to draw G with straight line edges. The nodes in V_1 and V_2 have to be placed on two parallel lines H_1 and H_2 such that the number of edge crossings is minimal. Assume for a moment that the nodes on the H_1 are fixed, and only the nodes on H_2 are permuted. For each pair of nodes on H_2, we introduce a variable x_{uv} such that $x_{uv} = 1$ if u is drawn to the left of v and $x_{uv} = 0$ otherwise. For edges (i, k) and (j, l) with $i, j \in H_1$ and $k, l \in H_2$, such that i is left of j, a crossing exists if and only if l is left of k. We thus have to punish x_{lk} in the objective function. The task of minimizing the number of crossings is now equivalent to determining a minimum linear ordering on the nodes of H_2. Note that bipartite crossing minimization with one fixed layer is already NP-hard [3]. If the nodes on both layers are allowed to permute, the problem can be modeled as a quadratic optimization problem over linear ordering variables. The quadratic linear ordering problem (QLO) is

$$
\begin{array}{lll}
 & \min & \sum_{(i,j,k,l)\in I} c_{ijkl} x_{ij} x_{kl} \\
(QLO) & \text{s.t.} & x \in P_{LO} \\
 & & x_{ij} \in \{0, 1\} \ \forall (i, j) \in J
\end{array}
$$

where P_{LO} is the linear ordering polytope and

$$I = \{(i,j,k,l) \mid i,j \in H_1,\ i < j,\ \text{and } k,l \in H_2,\ k < l\}$$
$$J = \{(i,j) \mid i,j \in H_1 \text{ or } i,j \in H_2, i < j\}$$

We replace each product $x_{ij}x_{kl}$ by a new binary variable y_{ijkl} and add the linearization constraints $y_{ijkl} \leq x_{ij}, y_{ijkl} \leq x_{kl}, y_{ijkl} \geq x_{ij} + x_{kl} - 1$. We call the resulting linearized problem (LQLO). In [1] it was shown that a 0/1 vector (x, y) satisfying $y_{ijkl} = x_{ij}x_{kl}$ is feasible for (LQLO) if and only if

$$x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} = 0 \ \forall (i,j,k,l) \in I, \tag{1}$$

which is a quadratic reformulation of the constraints defining P_{LO}. Note that (LQLO) is a quadratic binary optimization problem where the feasible solutions need to satisfy further side constraints. As unconstrained binary quadratic optimization is equivalent to the maximum cut problem [2], the task is to intersect a cut polytope with a set of hyperplanes. [1] showed that the hyperplanes (1) cut out faces of the cut polytope.

Let us consider a triple of leaves a, b, c in one of the trees. In case all pairwise lowest common ancestors coincide, all relative orderings between a, b, and c are feasible. However, if the lowest common ancestor of, say, a and b is on a lower level than that of, say, a and c, then c must not be placed between a and b; see Figure 1.

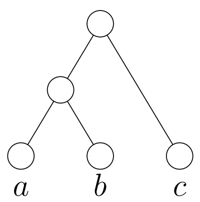

Fig. 1. Leaf c is not allowed to lie between a and b.

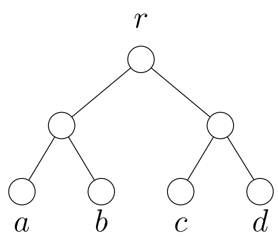

Fig. 2. Variables x_{ac} and x_{bd} can be identified.

Therefore, we derive a betweenness restriction for every triple of leaves such that two of the leaf pairs have different lowest common ancestors. Each restriction of the form 'c cannot be placed between a and b' can be written as $x_{ac}x_{cb} = 0$ and $x_{bc}x_{ca} = 0$, i.e., $y_{accb} = 0$ and $y_{cabc} = 0$. As mentioned above, the polytope corresponding to (LQLO) is isomorphic to a face of a cut polytope [1]. Since all y-variables are binary, the additional betweenness constraints are always face-inducing for (LQLO).

Theorem 1 *The problem of drawing tanglegrams with a minimum number of edge crossings can be solved by optimizing over a face of a suitable cut polytope.*

4 Results

We implemented the above model. The naive approach is to solve the linearized model (LQLO) using a standard integer programming solver. A more advanced approach is to solve the quadratic reformulation (1), using separation of cutting planes for max-cut, both in the context of integer and semidefinite programming. For the IP-based methods, we used CPLEX 11.2, whereas for the SDP approaches, we used the bundle method by [6]. For random and real-world binary as well as general trees with low tangle density, the SDP approach usually needs considerably more time than the IP-based methods. Furthermore, memory requirements strongly increase with system size. On average, the fastest approaches are the pure standard linearization *IP* and the quadratic reformulation *QP*. In fact, we can optimize tanglegrams with more than 500 leaves in each tree which is the range of real-world instance sizes. For big enough tangle density the SDP approach usually outperforms the IP approaches. Memory requirements, however, usually prohibit solving binary instances with more than 500 leaf nodes and tangle density of 1%. For general trees, best performance is often found for the quadratic reformulation. These non-binary instances could not be solved before by any other exact method.

References

[1] C. Buchheim and A. Wiegele, and L. Zheng. Exact Algorithms for the Quadratic Linear Ordering Problem. *INFORMS Journal on Computing.* To appear.

[2] C. De Simone. The Cut Polytope and the Boolean Quadric Polytope. *Discrete Mathematics*, 79; 71–75, 1989.

[3] P. Eades and N. C. Wormald. Edge crossings in drawing bipartite graphs. *Algorithmica*, 11; 379–403, 1994.

[4] H. Fernau and M. Kaufmann, and M. Poths. Comparing trees via crossing minimization. *Journal of Computer and System Sciences*. In Press.

[5] M. Nöllenburg, M. Völker, A. Wolff, and D. Holten. Drawing Binary Tanglegrams: An Experimental Evaluation. In *Proc. of the Workshop on Algorithm Engineering and Experiments, ALENEX 2009*, pages 106-119, SIAM, 2009.

[6] F. Rendl, G. Rinaldi, and A. Wiegele. A Branch and Bound Algorithm for Max-Cut Based on Combining Semidefinite and Polyhedral Relaxations. In M. Fischetti and D. P. Williamson, editors, *IPCO 2007*, volume 4513 of *Lecture Notes in Computer Science*, pages 295-309. Springer, 2007.

[7] B. Venkatachalam, J. Apple, K. St. John, and D. Gusfield. Untangling Tanglegrams: Comparing Trees by Their Drawings. *Bioinformatics Research and Applications*, pages 88–99, 2009.