A Satisfiability-based Approach for Embedding Generalized Tanglegrams on Level Graphs

Ewald Speckenmeyer[1], Andreas Wotzlaw[1⋆], and Stefan Porschen[2]

[1] Institut für Informatik, Universität zu Köln, Pohligstr. 1, D-50969 Köln, Germany
{esp, wotzlaw}@informatik.uni-koeln.de
[2] Fachbereich 4, HTW-Berlin, Treskowallee 8, D-10318 Berlin, Germany
porschen@htw-berlin.de

Abstract. A tanglegram is a pair of trees on the same set of leaves with matching leaves in the two trees joined by an edge. Tanglegrams are widely used in computational biology to compare evolutionary histories of species. In this paper we present a formulation of two related combinatorial embedding problems concerning tanglegrams in terms of CNF-formulas. The first problem is known as planar embedding and the second as crossing minimization problem. We show that our satisfiability formulation of these problems can handle a much more general case with more than two, not necessarily binary or complete, trees defined on arbitrary sets of leaves and allowed to vary their layouts.

Keywords: satisfiability, mixed Horn formula, 2-CNF, level graph, tanglegram, planar embedding, crossing minimization, graph drawing.

1 Introduction

In this paper we are interested in two combinatorial embedding problems concerning *generalized tanglegrams on level graphs*, a generalization of the well-known binary tanglegrams. A *binary tanglegram* [17] is an embedding (drawing) in the plane of a pair of rooted binary trees whose leaf sets are in one-to-one correspondence (perfect matching), such that matching leaves are connected by inter-tree edges. Clearly, the number of crossings between the inter-tree edges depends on the layout of the trees. From a practical point of view, an embedding with many crossings can hardly be analyzed. Fig. 1 shows an example of a binary tanglegram coming from phylogenetic studies done by Charleston and Perkins [5]. Thus, the first problem one can consider here consists of determining an embedding of one or both trees such that the inter-tree edges do not cross, if such an embedding exists. This problem is known as the *planar embedding* problem. If such a planar embedding is not possible, then we may want to find an embedding with as few crossing inter-tree edges as possible. This second problem, *crossing minimization*, is known in the literature also as the *tanglegram layout* problem [2, 3, 22].

⋆ The second author was partially supported by the DFG project under the grant No. 317/7-2

Both problems are motivated by the desire to find a good display of hierarchical structures, e.g., in software engineering, project management, or database design. They belong to the area of graph drawing [7]. Matching and aligning trees is also a recurrent problem in computational biology [17]. Embeddings with fewer crossings or with matching leaves close together are useful in biological analysis [22]. An embedding imposes an order among the leaves of the tree. Therefore, comparing the drawings of the trees is equivalent to comparing the permutations of the leaves. Two prominent applications are the comparison of phylogenetic trees [5, 6, 8] and the comparison of RNA structures [14, 20].

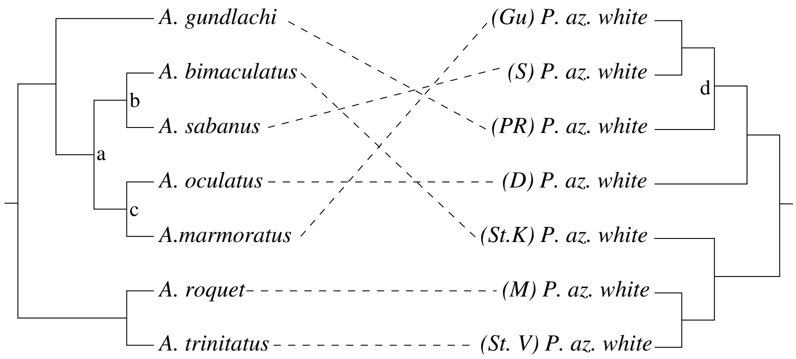

Fig. 1. A binary tanglegram from [5] showing phylogenetic trees for lizards (left tree) and strains of malaria (right tree) found in the Caribbean tropics. The dashed lines represent the host-parasite relationship. Here, the number of crossings is 7. This can be reduced to 1 by interchanging the children of nodes a, b, c, and d.

Bansal et al. [2] analyzed *generalized tanglegrams* where the number of leaves in the two binary trees may be different and a leaf in one tree may match multiple leaves in the other tree, thus no perfect matching is required here. They pointed out that such a generalization of the problem makes it possible to address not only the gene tree and species tree embedding problem, but also those problems in which the inter-tree edges between the trees can be completely arbitrary. Such general instances arise in several settings, e.g., in the analysis of host-parasite cospeciation [17].

Crossing minimization in tanglegrams has parallels to crossing minimization in graphs. Computing the minimum number of crossings in a graph is NP-complete [12]. However, it can be verified in linear time that a graph has a planar embedding [13]. The last assertion holds also for a more special case of level graphs [16, 19]. Computing the minimum number of crossings is fixed-parameter tractable [3, 15]. Analogously, crossing minimization in tanglegrams is NP-complete, as shown by Fernau et al. [10] by a reduction from the MAX-CUT problem [11], while the special case of planarity test can be decided in linear time [10]. Furthermore, the problem of minimizing the number of crossings where one tree is fixed and the layout of the other tree is allowed to vary can be solved efficiently. For binary trees with arbitrary topology, Fernau et al. [10] showed an $O(n \log^2(n))$ solution, further improved to $O(n \log^2(n)/\log\log(n))$ by Bansal et al. [2]. Here, n gives the number of leaves in each tree. Venkatachalam et al. [22] provided recently an algorithm working on the integer programming formulation of the problem with the so far best-known time bound of $O(n \log(n))$. For the case of generalized tanglegrams, Bansal et al. [2] presented two algorithms with running times $O(m \log^2(m)/\log\log(m))$ and $O(mh)$, where m is the number of edges between the two trees and h is the height of the tree whose layout can change. Based on the result of Fernau et al. [10], they also showed that the existence of planar embedding can be verified in $O(m)$ time.

In our generalization of the tanglegram problem we go even further than Bansal et al. [2]. In *generalized tanglegrams on level graphs* we consider problem instances with more than two trees where every tree is defined on an arbitrary set of leaves. Notice that here the pairwise disjoint leaf sets and the corresponding inter-tree edges (no perfect matching) connecting two neighboring leaf sets constitute a level graph [19] where each level is defined by some leaf set. Thus, each tree defined on some level implies additional constraints reducing considerably the set of possible embeddings. E.g., k-ary trees with n leaves allow for at most $k!^{\frac{n-1}{k-1}}$ different leaf orders implied by different orderings of the subtrees, i.e., 2^{n-1} in case of binary trees, compared with $n!$ permutations if no restrictions are imposed on the order of the leaves. Furthermore, in our setting we do not restrict the tanglegrams only to binary trees.

In this paper we present formulations of the planarity test and the crossing minimization problem on generalized tanglegrams on level graphs in terms of CNF-formulas by incorporating the ideas used already for level graphs in [19, 21]. By doing this, the planarity test essentially reduces to testing satisfiability of a 2-CNF formula. The crossing minimization problem has a formulation as a PARTIAL MAX-SAT problem of a CNF formula with a mandatory part of 3- and 2-clauses that must be satisfied for the solution to be reasonable, and a second part of 2-clauses such that its truth assignment must satisfy as many of these clauses as possible. In the mandatory part, the 3-clauses reflect transitivity conditions forced by the genus of the surface, whereas the 2-clauses reflect antisymmetry conditions. These clauses have to be satisfied in order to obtain a layout. The second part of 2-clauses reflects non-crossing conditions. Each unsatisfied clause from this part represents one arc crossing. This formulation offers a simple alternative for finding reasonable approximate solutions for the crossing minimization problem. We show that the planarity test for a generalized tanglegram on a level graph having a total of n vertices and with k-ary trees defined on each level, for some fixed integer $k > 1$, can be solved in $O(n^2)$ time by an elementary 2-SAT algorithm. Finally, to the best of our knowledge, this is the first time that the generalized tanglegram problem has been treated by means of a satisfiability formulation.

The rest of the paper is organized as follows. In Section 2 we provide some basic notation and definitions of relevant computational problems for generalized tanglegrams on level graphs. The satisfiability-based formulation of the two main problems on generalized tanglegrams on level graphs is given in Section 3. Finally, in Section 4 we conclude our paper and state some open questions.

2 Preliminaries and basic notation

Formally, a *level graph* is a triple (G, λ, L) where $G = (V, E)$ is a directed graph, $L = \{1, ..., |L|\}$ is the set of levels, and $\lambda : V \to L$ is the level-mapping, that assigns the vertices to levels such that each arc is directed from a lower to a higher level, i.e., $\forall e = (u, v) : \lambda(v) > \lambda(u)$. For simplicity, we identify the above triple by G having the other two components in mind. Observe that there exists no arc between vertices on the same level. If in addition, for every arc $e = (u, v) \in E, \lambda(v) = \lambda(u) + 1$ holds, then the level graph is called *proper*. In the present paper we consider proper level graphs only, hence we simply will speak of level graphs. This restriction means no loss of generality since an arbitrary level graph can be turned into a proper one preserving the crossing number by simply adding dummy vertices as shown in [9, 19].

Level graphs are drawn in the Euclidean x, y-plane by linear order, i.e., all vertices on the same level $j \in L$ are placed at arbitrary different positions on the line $y = j$; the x-coordinate of vertex u is denoted as $x(u)$. Arcs are represented by straight lines between the points representing their incident vertices. Often arrows at arc heads are omitted since the direction is implicitly fixed by the levels. For two vertices u, v on the same level, we simply write $u < v$ iff $x(u) < x(v)$. One is especially interested in level-graph drawings such that no two arc lines cross outside their endpoints. A level graph for which such a drawing exists is called *level-planar*. It is not hard to see that a level graph with $|E| > 2|V| - 4$ cannot be level-planar [19]. Therefore, for most level graphs all what one can hope for is to find a plane embedding such that the number of arc-crossings is minimized. Moreover, by reduction from the FEEDBACK ARC SET problem [11], Eades and Wormald [9] showed that crossing minimization in level graphs is NP-hard, even if there are only two levels with a fixed order of nodes on one level.

In generalized tanglegrams on level graphs, we define additionally on the nodes of each level $i \in L$ of a level graph G a tree T_i with nodes of level i as leaf set. Clearly, the presence of a tree on each level reduces the search space of admissible embeddings considerably. More formally, a generalized tanglegram on a level graph G is a quadruple (G, λ, L, F) where $F = \{T_1, ..., T_{|L|}\}$ is a forest of *level-trees* and G, λ, and L are defined as above. We say that a rooted level-tree is *complete* if all its leaves have the same depth. Given a rooted, unordered tree $T \in F$, we write $V(T)$, and $E(T)$ to denote its node set, and edge set, respectively. Furthermore, for two trees T_i and T_{i+1} from F defined on two adjacent levels i and $i+1$ of level graph G, we define the set of *inter-tree arcs* as

$$E(T_i, T_{i+1}) := \{(u, v) \in E(G) : \lambda(u) = i, \lambda(v) = i + 1\}.$$

Observe that for a proper graph G holds $E(G) = \bigcup_{i=1,...,|L|-1} E(T_i, T_{i+1})$.

For each node $v \in V(T)$, let $T(v)$ denote the subtree of T rooted at v. Given a tree T, we say that a linear order σ on the leaves of T is *compatible* with T if for each node $v \in V(T)$ the leaves in $T(v)$ form an interval (i.e., appear as a consecutive block) in σ. We write $u <_\sigma v$ to mean that leaf u appears before leaf v in the linear order σ on the leaves of T. Given compatible linear orders σ_i and σ_{i+1} on two trees T_i and T_{i+1} from F defined on two adjacent levels i and $i+1$ of level graph G, respectively, the *number of crossings* between σ_i and σ_{i+1} among the inter-tree arcs $E(T_i, T_{i+1})$ is defined as

$$\tau(\sigma_i, \sigma_{i+1}) := \Big|\Big\{\{(u,a),(v,b)\} \subseteq E(T_i, T_{i+1}) : \neg\big((u <_{\sigma_i} v) \leftrightarrow (a <_{\sigma_{i+1}} b)\big)\Big\}\Big|.$$

Note that a pair of arcs cross at most once (see Fig. 1). Moreover, since we assume here that G is a proper level graph, only adjacent levels can induce crossings. Finally, the overall number of crossings for an instance (G, λ, L, F) and a set $S := \{\sigma_1, ..., \sigma_{|L|}\}$ of compatible orders for each level in L (tree in F) is defined as

$$\tau(G, \lambda, L, F, S) := \sum_{i=1,...,|L|-1} \tau(\sigma_i, \sigma_{i+1}).$$

Problem 1 (Planarity Test). Given an instance (G, λ, L, F), verify if there exists a planar embedding, i.e., if there exists some set S of compatible linear orders σ_i for each level $i \in L$ (tree $T_i \in F$) such that $\tau(G, \lambda, L, F, S) = 0$.

Problem 2 (Crossing Minimization). Given an instance (G, λ, L, F), find a set S of compatible linear orders σ_i for each level $i \in L$ (tree $T_i \in F$) such that $\tau(G, \lambda, L, F, S)$ is minimized.

To complete the notation, let CNF denote the set of formulas (free of duplicate clauses) in conjunctive normal form over a set $V = \{x_1, ..., x_n\}$ of propositional variables $x_i \in \{0, 1\}$. Each variable x induces a positive literal (variable x) or a negative literal (negated variable $\overline{x}$). Each formula $C \in$ CNF is considered as a clause set $C = \{c_1, ..., c_{|C|}\}$. Each clause $c \in C$ is a disjunction of different literals l_i, and is also represented as a set $c = \{l_1, ..., l_{|c|}\}$. A clause is termed a k-clause, for some $k \in \mathbb{N}$, if it contains at most k literals. The number of clauses in C is denoted by $|C|$. For $k \in \mathbb{N}$, let k-CNF denote the subset of formulas C such that each clause has length at most k. We denote by $V(C)$ the set of variables occurring in formula C. The satisfiability problem (SAT) asks, whether formula C is *satisfiable*, i.e., whether there is a truth assignment $t : V(C) \to \{0, 1\}$ setting at least one literal in each clause of C to 1. Given $C \in$ CNF, the optimization version MAX-SAT searches for a truth assignment t satisfying as many clauses of C as possible.

3 Satisfiability formulation of crossing minimization

In the following we provide a formulation of the crossing minimization problem for generalized tanglegrams on level graphs in terms of propositional logic.

We proceed in two steps. Given a generalized tanglegram (G, λ, L, F), we first show the construction of CNF-formulas for the level graph (G, λ, L). In the second step, we describe a similar construction for the forest F of the generalized tanglegram.

Consider in a proper level graph G two subsequent levels i and $i+1$ from L, as shown in Fig. 2. Let $e = (u, a)$ and $f = (v, b)$ be two arcs from $E(T_i, T_{i+1})$ directed from level i to level $i+1$ with different tails $u \neq v$ and different heads $a \neq b$. In a drawing of G, e and f do not cross iff

$$u <_\sigma v \quad \Leftrightarrow \quad a <_\sigma b$$

for some linear order σ. Observe that arcs having the same head or tail never cross in any drawing of G.

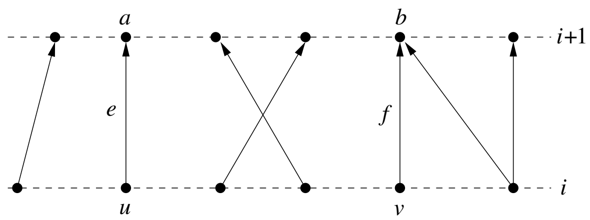

Fig. 2. Adjacent levels i and $i+1$ of a level graph G. Arcs $e = (u, a)$ and $f = (v, b)$ have different tails and heads.

The construction of a Boolean formula C_G representing the plane embedding of G proceeds as follows:

1. For each level $i \in L$ and every pair $\{u, v\}$ of distinct vertices from level i, i.e., $\lambda(u) = \lambda(v) = i$, create a Boolean variable uv that is true iff $u <_\sigma v$ for some linear order σ.
2. Create the following Boolean subformulas:
   (i) For each level $i \in \{1, ..., |L| - 1\}$ and every two arcs $e = (u, a), f = (v, b)$ from $E(T_i, T_{i+1})$ having their tails $u \neq v$ on level i and heads $a \neq b$ on level $i+1$, form the non-crossing preserving expression:

$$uv \leftrightarrow ab$$

   (ii) For each level $i \in \{1, ..., |L|\}$ and each pair $\{u, v\}$ of distinct vertices on level i, form the antisymmetry expression:

$$uv \leftrightarrow \overline{vu}$$

   (iii) For each level $i \in \{1, ..., |L|\}$ and each triple $\{u, v, w\}$ of distinct vertices on level i, form the transitivity expression:

$$uv \wedge vw \rightarrow uw$$

Observe that the formulas resulting from (i) and (ii) yield 2-CNF formulas C_i and C_{ii} via

$$a \leftrightarrow b \equiv (\overline{a} \vee b) \wedge (\overline{b} \vee a).$$

The formula resulting from (iii) yields a Horn formula C_{iii} with clauses of length 3 via elementary equivalence

$$(a \wedge b \rightarrow c) \equiv (\overline{a} \vee \overline{b} \vee c).$$

Recall that each clause of a Horn formula contains at most one positive literal. Hence the formula $C_G = C_i \wedge C_{ii} \wedge C_{iii}$ encoding the plane embedding of a level graph G is a mixed Horn formula [18]. If G has n vertices distributed over $|L|$ levels then C_G has $|V(C_G)| \in O(n^2)$ variables. Moreover, by counting $|C_i| \in O(|E(G)|^2)$, $|C_{ii}| \in O(n^2)$, and $|C_{iii}| \in O(n^3)$. Hence the number of clauses in C_G is bounded by $O(n^3 + |E(G)|^2)$. As mentioned before, the maximal number of arcs in a level-planar graph containing $n > 2$ nodes is at most $2n - 4$. Thus, in the case we use C_G for a level planarity test, a preprocessing ensures that only $O(n^2)$ 2-clauses in C_i are generated. The following result shows that the level planarity test can be formulated as a satisfiability problem.

Proposition 1 ([19]). *A level graph G with n vertices has a level-planar embedding iff $C_G - C_{iii}$ is satisfiable. The test can be done in time $O(n^2)$.*

According to [19], the transitivity formula C_{iii} is superfluous for the level planarity test. This results in a better complexity of $O(n^2)$, since SAT for 2-CNF formulas can be decided in linear time in the number of variables and clauses in the input formula [1].

Minimizing the number of crossings of G is equivalent in terms of propositional calculus to determining a truth assignment which satisfies all clauses in C_{ii} and C_{iii} and which maximizes the number of satisfied clauses in C_i. This optimization problem is known as PARTIAL MAX-SAT [4], a variant of the MAX-SAT problem, and remains NP-hard even for (unsatisfiable) 2-CNF instances. Unfortunately, it turns out that for considering crossing minimization in terms of PARTIAL MAX-SAT, formula C_{iii} cannot be dropped in general [21].

Proposition 2 ([19]). *Let G be a level graph and $t : V(C_G) \rightarrow \{0, 1\}$ be a truth assignment satisfying all clauses of C_{ii} and C_{iii} and minimizing the number τ_G of violated clauses in C_i. Then τ_G is the minimum number of arc crossings in a level embedding of G.*

Consider now some tree T_i from F built on a level i from L. Without loss of generality assume that T_i is a complete, k-ary tree of height d, for some integers $k, d > 1$. Note that for $d = 1$ the edges of T_i never cross in any drawing of T_i and the generation of a CNF formula C_{T_i} for T_i can be omitted. Let w be some node from $V(T_i)$ such that the height of subtree $T_i(w)$ is at least 2. Note that the edges of $T_i(w)$ connecting nodes of depth 0 and 1 never cross in any drawing of $T_i(w)$. Therefore, let $e = \{u, a\}$ and $f = \{v, b\}$ be two edges from $E(T_i(w))$

with $u \neq v$ having both depth 1 and $a \neq b$ being some children of u and v, respectively, as shown in Fig. 3. In a drawing of $T_i(w)$, e and f do not cross iff

$$u <_\sigma v \quad \Leftrightarrow \quad a <_\sigma b$$

for some linear order σ.

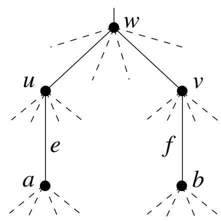

Fig. 3. Part of subtree $T_i(w)$ with two non-crossing edges e and f.

We describe now the construction of a Boolean formula C_{T_i} encoding the plane embedding of T_i. We proceed as follows:

1. For each level $j = 1, ..., d$ of T_i and every pair $\{u, v\}$ of distinct vertices from level j, create a Boolean variable uv that is true iff $u <_\sigma v$ for some linear order σ.
2. Create the following Boolean subformulas:
   (iv) For each level $j = 1, ..., d-1$ of T_i and every two edges $e = \{u, a\}$ and $f = \{v, b\}$ from $E(T_i)$ such that $u \neq v$ have depth j and a and b have depth $j+1$ in T_i, form the non-crossing preserving expression:

$$(uv \to ab) \wedge (vu \to ba)$$

   (v) For each level $j = 1, ..., d$ and each pair $\{u, v\}$ of distinct vertices of depth j in T_i, form the antisymmetry expression:

$$uv \leftrightarrow \overline{vu}$$

Notice that the formulas resulting from (iv) and (v) yield after some elementary transformations 2-CNF formulas $C_{iv}^{T_i}$ and $C_v^{T_i}$, respectively, for each tree T_i. We proceed with the generation of Boolean formulas $C_{T_i} = C_{iv}^{T_i} \wedge C_v^{T_i}$ for all trees from F and obtain finally a Boolean formula

$$C_F = \bigwedge_{T_i \in F} C_{T_i}$$

encoding the plane embedding of F.

We shall now estimate the length of each formula C_{T_i}. The number of variables generated for each level $j = 1, ..., d$ of a k-ary tree T_i is equal to $\binom{k^j}{2}$ and thus bounded by $O(k^{2j})$. If $r_i \leq n$ is the number of vertices in level $i \in L$ of graph G, then the height of any k-ary complete tree T_i is at most $\lceil \log_k(r_i) \rceil$. Hence, each C_{T_i} has $O\big(\frac{r_i^2-1}{k^2-1}\big)$ variables. Furthermore, the number of 2-clauses contributed to formula $C_{iv}^{T_i}$ by a level $j \in \{1, ..., \lceil \log_k(r_i) \rceil - 1\}$ of T_i is at most $2k^2\binom{k^j}{2} \in O(k^{2+2j})$, what summed up over $\lceil \log_k(r_i) \rceil - 1$ tree levels yields $|C_{iv}^{T_i}| \in O\big(\frac{r_i^2-k^2}{k^2-1}\big)$. For the number of clauses in $C_v^{T_i}$ we proceed similar as for the number of variables above and obtain that $|C_v^{T_i}| \in O\big(\frac{r_i^2-1}{k^2-1}\big)$. Thus, the number of 2-clauses in C_{T_i} is bounded by $O(r_i^2)$ for some fixed integer $k > 1$. Notice that in case of a tree T_i with r_i leaves but of height greater than $\lceil \log_k(r_i) \rceil$, there must be an inner node in $V(T_i)$ with less than k children. That yields formulas $C_{iv}^{T_i}$ and $C_v^{T_i}$ with less variables and clauses than for the case of the k-ary complete tree with r_i leaves. Similar to Proposition 1, we obtain finally the following result for T_i:

Proposition 3. *For some fixed integer $k > 1$, a k-ary tree T_i built on a level i with r_i vertices has a planar embedding iff C_{T_i} is satisfiable. The test can be done in time $O(r_i^2)$.*

Since r_i is the number of vertices on level $i \in L$ in graph G and $r_1 + ... + r_{|L|} = n$, it follows that $|V(C_F)| \in O(n^2)$ and $|C_F| \in O(n^2)$.

Corollary 1. *For some fixed integer $k > 1$, a set of k-ary trees built on a level graph G with n vertices has a planar embedding iff C_F is satisfiable. The test can be done in time $O(n^2)$.*

Note that every satisfying truth assignment for C_F induces compatible linear orders σ_i on the leaves of each $T_i \in F$, and vice versa.

We are now ready to give a final satisfiability-based formulation for an instance (G, λ, L, F) of a generalized tanglegram on a level graph G. To this end, we simply generate CNF formulas C_G and C_F for (G, λ, L) and F, respectively, as described above, and combine them into a new CNF formula as follows

$$C_{GF} = C_G \wedge C_F = (C_i \wedge C_{ii} \wedge C_{iii}) \wedge \bigwedge_{T_i \in F} \big(C_{iv}^{T_i} \wedge C_v^{T_i}\big).$$

For a level graph G with n vertices and k-ary trees F defined on its levels L, the number of clauses in C_{GF} is bounded by $O(n^3 + |E(G)|^2)$, according to the discussion above. Furthermore, C_{GF} has $O(n^2)$ variables. Note that these estimates hold only for some fixed integer $k > 1$.

Since C_{GF} contains 3-clauses, it cannot in general be solved for SAT efficiently. However, since the transitivity formula $C_{iii} \in$ 3-CNF is superfluous for the planarity test, we can remove it from C_{GF}, thus obtaining a 2-CNF formula. Similarly as for Proposition 1, we can now solve the planarity test for (G, λ, L, F) in time $O(n^2)$ by applying the algorithm of Aspvall et al. [1]. Recall that the maximal number of arcs in a level-planar graph containing $n > 2$ nodes is at most $2n - 4$. Hence, the number of clauses $|C_{GF} - C_{iii}| \in O(n^2)$.

Proposition 4. *Let (G, λ, L, F) be an instance of a generalized tanglegram on a level graph G with n vertices and k-ary trees F, for some fixed integer $k > 1$. Then (G, λ, L, F) has a planar embedding iff $C_{GF} - C_{iii}$ is satisfiable. The test can be done in time $O(n^2)$.*

Minimizing the number of crossings of (G, λ, L, F) is equivalent to determining a truth assignment which satisfies all clauses in $C_{GF} - C_i$ and which maximizes the number of satisfied clauses in C_i, thus solving an instance of the PARTIAL MAX-SAT problem. Again, for considering crossing minimization in terms of PARTIAL MAX-SAT, formula $C_{iii} \in$ 3-CNF cannot be dropped.

Proposition 5. *Let (G, λ, L, F) be an instance of a generalized tanglegram on a level graph G with n vertices and k-ary trees F, for some fixed integer $k > 1$, and let $t : V(C_{GF}) \to \{0, 1\}$ be a truth assignment satisfying all clauses of $C_{GF} - C_i$ and minimizing the number τ of violated clauses in C_i. Then τ is the minimum number of arc crossings in an embedding of (G, λ, L, F).*

Observe that compatible linear orders σ_i for each level $i \in L$ can be extracted from a truth assignment t in time $O(n^2)$ by traversing all variables of C_{GF}.

4 Conclusion and open problems

We have presented a satisfiability-based formulation of the planarity test and the crossing minimization problem on generalized tanglegrams defined on level graphs. Here, the first problem essentially reduces to testing satisfiability of a 2-CNF formula and can be solved in $O(n^2)$ time for instances with n level vertices and k-ary trees defined on each level, for some fixed integer $k > 1$. Moreover, we have shown that the latter problem has a formulation as a PARTIAL MAX-SAT problem. Here, the question arises whether one could derive bounds on the approximation ratio for generalized tanglegram instances. From a practical point of view, it would be interesting to test the efficiency of our satisfiability-based approach against other techniques while solving (generalized) binary tanglegrams.

References

1. Aspvall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified Boolean formulas. Information Processing Letters 8(3), 121–123 (1979)
2. Bansal, M.S., Chang, W., Eulenstein, O., Fernández-Baca, D.: Generalized binary tanglegrams: Algorithms and applications. In: Proceedings of the 1st International Conference on Bioinformatics and Computational Biology. Lecture Notes in Computer Science, vol. 5462, pp. 114–125 (2009)
3. Buchin, K., Buchin, M., Byrka, J., Nöllenburg, M., Okamoto, Y., Silveira, R.I., Wolff, A.: Drawing (complete) binary tanglegrams: Hardness, approximation, fixed-parameter tractability. In: Proceedings of the 16th International Symposium on Graph Drawing. Lecture Notes in Computer Science, vol. 5417, pp. 324–335 (2008)

4. Cha, B., Iwama, K., Kambayashi, Y., Miyazaki, S.: Local search algorithms for partial MAXSAT. In: Proceedings of the 14th National Conference on Artificial Intelligence (AAAI/IAAI). pp. 263–268 (1997)
5. Charleston, M.A., Parkins, S.L.: Lizards, malaria, and jungles in the Caribbean. In: Tangled Trees: Phylogeny, Cospeciation and Coevolution, pp. 65–92. University of Chicago Press (2003)
6. DasGupta, B., He, X., Jiang, T., Li, M., Tromp, J.: On the linear-cost subtree-transfer distance between phylogenetic trees. Algorithmica 25(2-3), 176–195 (1999)
7. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for Geometric Representations of Graphs. Prentice Hall (1998)
8. Dufayard, J., Duret, L., Penel, S., Gouy, M., Rechenmann, F., Perriere, G.: Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. Bioinformatics 21, 2596–2603 (2005)
9. Eades, P., Wormald, N.C.: Edge crossings in drawings of bipartite graphs. Algorithmica 11(4), 379–403 (1994)
10. Fernau, H., Kaufmann, M., Poths, M.: Comparing trees via crossing minimization. Journal of Computer and System Sciences 76(7), 593–608 (2010)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)
12. Garey, M.R., Johnson, D.S.: Crossing number is NP-complete. SIAM Journal on Algebraic and Discrete Methods 4(3), 312–316 (1983)
13. Hopcroft, J., Tarjan, R.E.: Efficient planarity testing. J. ACM 21(4), 549–568 (1974)
14. Jansson, J., Hieu, N.T., Sung, W.: Local gapped subforest alignment and its application in finding RNA structural motifs. Journal of Computational Biology 13(3), 712–718 (2006)
15. Kawarabayashi, K., Reed, B.A.: Computing crossing number in linear time. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC). pp. 382–390 (2007)
16. Leipert, S.: Level planarity testing and embedding in linear time. Ph.D. thesis, Institut für Informatik, Universität zu Köln (1998)
17. Page, R.D.M.: Tangled Trees: Phylogeny, Cospeciation, and Coevolution. University of Chicago Press (2002)
18. Porschen, S., Speckenmeyer, E.: Satisfiability of mixed Horn formulas. Discrete Applied Mathematics 155(11), 1408–1419 (2007)
19. Randerath, B., Speckenmeyer, E., Boros, E., Hammer, P.L., Kogan, A., Makino, K., Simeone, B., Cepek, O.: A satisfiability formulation of problems on level graphs. Electronic Notes in Discrete Mathematics 9, 269–277 (2001)
20. Shapiro, B.A., Zhang, K.: Comparing multiple RNA secondary structures using tree comparisons. Bioinformatics 6(4), 309–318 (1990)
21. Speckenmeyer, E., Porschen, S.: PARTIAL MAX-SAT of level graph (mixed-Horn)formulas. Studies in Logic 3(3), 24–43 (2010)
22. Venkatachalam, B., Apple, J., St. John, K., Gusfield, D.: Untangling tanglegrams: Comparing trees by their drawings. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(4), 588–597 (2010)