Generalizing Tanglegrams

Balaji Venkatachalam
Google Inc.
ijalabv@gmail.com

Dan Gusfield
Department of Computer Science
UC Davis
dmgusfield@ucdavis.edu

26^{th} July, 2018

Abstract

Tanglegrams are a tool to infer joint evolution of species. Tanglegrams are widely used in ecology to study joint evolution history of parasitic or symbiotically linked species. Visually, a tanglegram is a pair of evolutionary trees drawn with the leaves facing at each other. One species at the leaf of one trees is related ecologically to a species at a leaf of another tree. Related species from the two trees are connected by an edge. The number of crossings between the edges joining the leaves indicate the relatedness of the trees. Earlier work on tanglegrams considered the same number of leaves on both the trees and one edge between the leaves of the two trees. In this paper we consider multiple edges from a leaf in the trees. These edges correspond to ecological events like duplication, host switching etc. We generalize the definition of tanglegrams to admit multiple edges between the leaves. We show integer programs for optimizing the number of crossings. The integer program has an XOR formulation very similar to the formulation for the tanglegrams. We also show how the ideas for distance minimization on tanglegrams can be extended for the generalized tanglegrams. We show that the tanglegram drawings used in ecology can be improved to have fewer crossings using our integer programs.

1 Introduction

Tanglegrams [**?**] have been used as a tool, primary in ecology, to study co-evolution, horizontal gene transfer, host-parasite interactions, mutualism. Tanglegrams are a visual way of inferring how different the evolutionary trees of two species under consideration are.

In earlier works [**?**, **?**] tanglegrams have been considered an equal number of leaves in both the trees and a one-one mapping between the leaves. In this work allow multiple edges from a leaf in one tree to multiple leaves in the other tree.

Biological motivation In *Coevolution of Life on Hosts*, Clayton et al. [**?**] show that there are coevolution phenomena like "duplication", "host switching", "cohesion" etc., where a host or a parasite can be linked to multiple parasites or hosts, respectively. For example, one species of a parasite might be dependent on multiple hosts due to host switching.

Generalizing Tanglegrams Tanglegrams considered in earlier works have exactly one edge from the leaf of one tree to another. That is, leaf i in the first tree is connected to leaf labeled i in the second tree. Let us call these as *match edges*. The generalized tanglegram includes edges between i of the first tree and j of the second tree, for $i \neq j$. Let us call these as *switch edges*.

The crossing minimization problem now is the minimize the crossings between all the leaf edges. The goal of distance minimization is to minimize the sum of the distances for all the lead edges – match edges and switch edges.

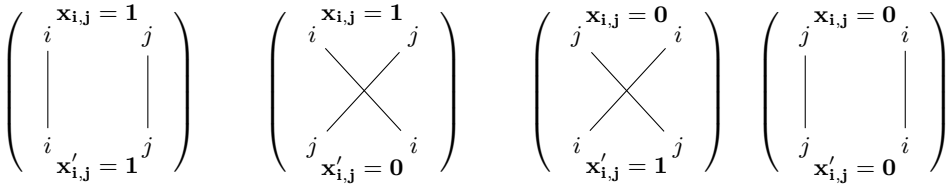
In the next section, we describe integer programs for crossing minimization. In the following section we describe integer programs for distance minimization. We describe integer programs for both the DP formulation and the related distance formulation. We then show that the formulation can improve the crossings in the drawings used in ecology, and shown the book by Clayton et al. [**?**].

2 Integer Linear Program for Crossing minimization

The formulation for crossing minimization is based on the following intuition: if the leaf i is to the left of leaf j in both of the trees, then the edges connecting the is and the js do not cross. The edges cross if there is an inversion in the order. To realize this, for the first tree, we introduce binary variables $x_{i,j}$ for all leaf pairs (i, j) such that $i < j$. $x_{i,j}$ is set to 1 iff i appears before j in the linear order. For every internal node k, we introduce a variable y_k. Let c_1 and c_2 be the two children of k. In a layout y_k is set to 1 if c_1, c_2 are placed to the left and right, respectively, otherwise $y_k = 0$. For all leaves i in the subtree below c_1 and j in the subtree below c_2, if $i < j$ then $x_{i,j} = 1 \iff y_k = 1$, i.e., $x_{i,j} = y_k$. If $j < i$, then $y_k = 1 - x_{i,j}$. Analogously, for the second tree, we define these constraints over variables $x'_{i,j}$ and y'_k.

2.1 Crossing between match edges

If i is to the left (or right) of j in the drawing of both trees in the tanglegram, then there is no crossing.



i and j cross only when the order is reversed. That is, (i, j) cross iff $x_{i,j} \neq x'_{i,j}$. We let $z_{i,j} = x_{i,j} \oplus x'_{i,j}$. We can rewrite the XOR as the following linear inequalities: $z_{i,j}$ - $x_{i,j}$ + $x_{i,j} \geq 0$; $z_{i,j}$ + $x_{i,j}$ - $x_{i,j} \geq 0$; $z_{i,j}$ - $x_{i,j}$ - $x_{i,j} \leq 0$; $z_{i,j}$ + $x_{i,j}$ + $x_{i,j} \leq 2$.

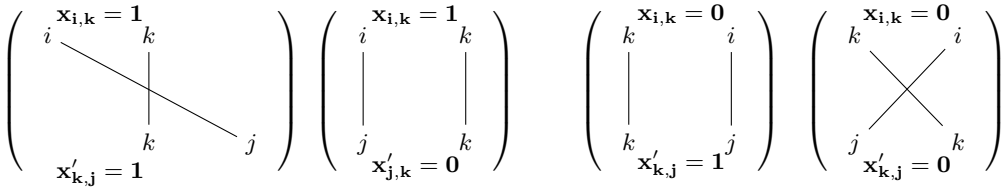
The objective function for minimizing the number of crossings between the match edges is, therefore, $min \Sigma_{i<j} z_{i,j}$.

2.2 Crossing between switch edges and match edges

We will first consider the case of switch edges crossing match edges. That is, switch edge (i, j) crossing with match edge (k, k). We will define a variable $z_{i,j,k}$ to denote the switch edge (i, j) crossing match edge (k, k). Notice that the variable name is well defined — there are no clashes between the variable names. The first variable of the subscript is the switch edge leaf in T_1, the second is the leaf of the second tree T_2 and the third variable is the match edge for all $k \in [n]$ i, j.

We will first consider all possible k for $i < j$. Then we will consider all cases of k for $j < i$.

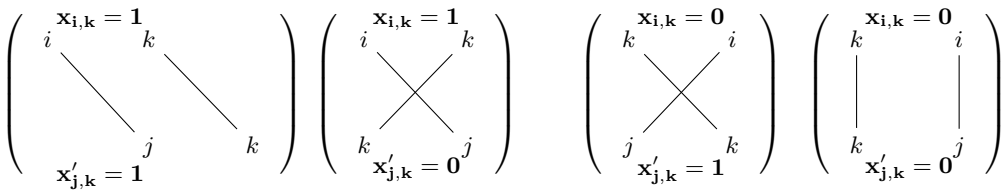
Case 1: $i < k < j$ We will first consider the case of k being between i and j. Since we define $x_{i,j}$ variables only for $i < j$, we have $x_{i,k}$ and $x'_{k,j}$. There are four cases, for every combination of the values being 0 or 1 for these variables. In each case, i, j, k denote the positions of the leaves on the two trees. The corresponding values of $x_{i,k}$ and $x'_i k, j$ are shown.



From these pictures $z_{i,j,k}$ is the XNOR of $x_{i,k}$ and $x_{k,j}$.

$$z_{i,j,k} = \neg(x_{i,k} \oplus x'_{k,j})$$

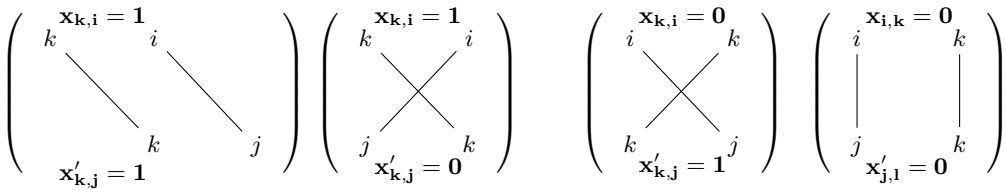
Case 2: $i < j < k$ We will next consider the case of k being bigger than both j and j.



This has the XOR relationship between $x_{i,k}$ and $x'_{j,k}$.

$$z_{i,j,k} = x_{i,k} \oplus x'_{j,k}$$

Case 3: $k < i < j$ We will finally consider the case of k being smaller than both j and j.

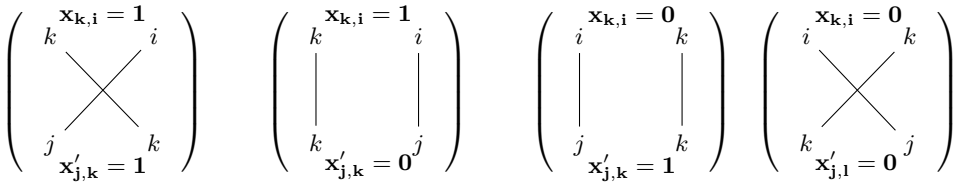


Again, like in Case 2, there is an XOR relationship between $x_{k,i}$ and $x'_{k,j}$.

$$z_{i,j,k} = x_{k,i} \oplus x'_{k,j}$$

We will next consider the case of $j < i$.

Case 4: $j < k < i$ We will first consider the case of k being between i and j.



From these pictures $z_{i,j,k}$ is the XNOR of $x_{i,k}$ and $x_{k,j}$.

$$z_{i,j,k} = \neg(x_{k,i} \oplus x'_{k,j})$$

For the other two cases $k < j < i$ is similar to $k < i < j$ (case 2 above) and $j < i < k$ is similar to $i < j < k$ (case 3 above). That is, $z_{i,j,k}$ has an XOR relationship between the x and the x' variables.

Note We are forced to use XNOR in a few cases because we have defined $x_{i,j}$ only for $i < j$. Had we also introduced $x_{j,i} = \neg x_{i,j}$, then we can simplify all the above cases to $z_{i,j,k} = x_{i,k} \oplus x_{j,k}$.

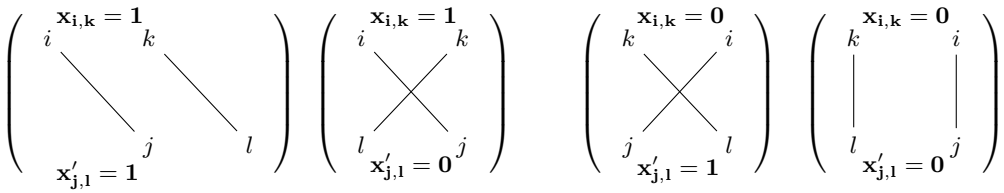
2.3 Crossing between switch edges

We will next consider the case of crossings between switch edges. Consider switch edges (i, j) and (k, l), where i and k are on tree T_1 and j, l are in T_2. Since we define $x_{i,j}$ variables only for $i < j$, we have $x_{i,k}$ and $x'_{j,l}$.

For switch edges (i, j) and (k, l), without loss of generality let $i < k$. That is, we consider the edges in lexicographically smaller pair as the first edge. We create a node $z_{i,j,k,l}$ for this pair of edges. Again, this is well defined and there are no clashes in variable names.

There are two cases, $j < l$ and $j > l$.

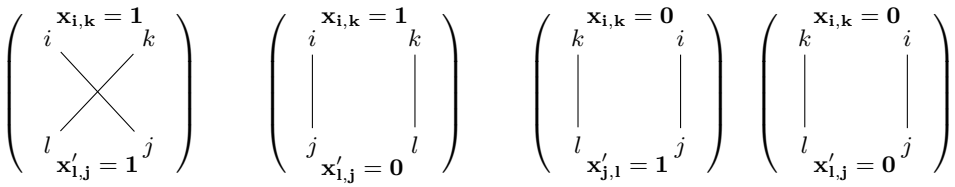
Case 1: $i < k$ **and** $j < l$ In the figures below all the configurations are considered. i, j, k, l denote the positions of the leaves on the two trees. The corresponding values of $x_{i,k}$ and $x'_i j, l$ are shown.



From these pictures $z_{i,j,k,l}$ is the XOR of $x_{i,k}$ and $x'_{j,l}$.

$$z_{i,j,k,l} = x_{i,k} \oplus x'_{j,l}$$

Case 2: $i < k$ **and** $l < j$ All configurations for this case are below.



From these pictures $z_{i,j,k}$ is the XNOR of $x_{i,k}$ and $x_{l,j}$.

$$z_{i,j,k,l} = \neg(x_{i,k} \oplus x'_{l,j})$$

Note Similar to the note of the previous subsection, had introduced both $x_{i,j}$ and $x_{j,i}$, then $z_{i,j,k,l} = x_{i,k} \oplus x'_{j,l}$.

2.4 Objective function

The number of crossings is the sum of the crossings of the match edges together with the crossings between the switch edges and the match edges and the crossings between the switch edges. Thus the objective function is:

$$\min \sum_{i<j} z_{i,j} + \sum_{(i,j)\in \text{switch}} z_{i,j,k} + \sum_{(i,j),(k,l)\in \text{switch}} z_{i,j,k,l}$$

3 Distance minimzation

We describe two different formulations for the distance minimization problem. The first formulation is based on the dynamic programming idea used in the one-tree distance minimization problem. The second uses the simple fact that the order of its children in an internal node determines the relation between the leaves in the two subtrees.

3.1 Dynamic Programming Formulation

In the dynamic programming algorithm for one-tree distance minimization (Section 3.2 of the journal paper), every subtree is rooted at every possible position so that its leaves are located starting at position i for all $i \in [n]$. Here, we will generate equations to allow placing each subtree of either tree at every position. The constraints will eliminate mutually incompatible configurations. The sum of the distances of the matching edges can be calculated for each legal layout of the trees. The objective is to determine the optimal solution among all possible layouts.

For a vertex k, we set a binary variable $y_{k,p} = 1$ when the subtree beneath it is placed starting at position p. For instance, $y_{\text{root},1} = 1$ always. If k is an internal node, let i and j be its children with l and r leaves in the subtrees below them. Either i is placed as the left child or the right child. If i is placed

as the left child then its leaves take positions p through $p+l-1$ and the leaves below j take positions $p+l$ through $p+l+r-1$.

$y_{k,p}=1$ implies that the node i is placed at position p or $p+r$. This implication is written by the inequality $y_{i,p}+y_{i,p+r}\geq y_{k,p}$. Similarly, $y_{j,p}+y_{j,p+l}\geq y_{k,p}$. Both i and j cannot be the left (or right) child of k simultaneously, so $y_{j,p}+y_{i,p}\leq 1$.

Every leaf must occur exactly once. For every leaf l, therefore, $\sum_{r\in[n]} y_{l,r}=1$. Every position must have exactly one leaf, so $r\in[n]$, $\sum_{l\in leaves} y_{l,r}=1$. We use variables y' and similar inequalities for the second tree.

To calculate the distance contributed by each leaf, we introduce the variable $z_{l,r,r'}$. Binary variables $zl,r,r'=1$ only when the leaf l is present at positions r,r' in the two trees, respectively. $z_{l,r,r'}$ contributes $|r-r'|$ to the distance value.

Switch edges To account for the distance contributed by the switch edges, we will introduce variables $z_{u,v,r,r'}$ for every switch edge (u,v). $z_{u,v,r,r'}=1$ only when u in T_1 is at position r and v is at position r' in T_2.

Therefore, the objective function is:

$$\min \sum_{\text{leaf}l} \sum_{r\in[n]} \sum_{r'\in[n]} |r-r'|z_{l,r,r'} + \sum_{(u,v)\in\text{switch}} \sum_{r\in[n]} \sum_{r'\in[n]} |r-r'|z_{u,v,r,r'}$$

3.2 Related Distance Formulation

In this formulation, we will use the relative distance between the pair of leaves on the same tree. This distance is determined by the order of the children at the least common ancestor.

Consider an internal node i with m leaves in its subtree and let its two children be c_1,c_2. Let j,k be leaves in subtrees c_1,c_2, respectively. Let x_j denote the position of leaf j in the linear order, $[n]$. Introduce a binary variable y_i for each internal node i to model the choice of c_1 or c_2 being the left child. $y_i=1$ when c_1 is the left child (and j is to the left of k). The opposite is implied by $y_i=0$. Now the order of the children c_1,c_2 determine the distance between the leaves in its subtrees.

$$y_i=1 \iff -(m-1)\leq x_j-x_k\leq -1 y_i=0 \iff 1\leq x_j-x_k\leq m-1 \tag{1}$$

These implications are written as the following inequal- ities: $x_j-xk+1\leq m(1-y_i)$ and $x_j-xk+my_i\geq 1$.

Next, we need to ensure that all leaves $1\leq x_j\leq n$ and all x_js are unique. The uniqueness constraints can be written in a number of ways. We model them as a matching problem. It has been observed in the ILP literature that the vertices of the matching polytope are all lattice points and, therefore, the ILP software need not apply further reduction techniques [25]. As usual, we define similar inequalities on variables x'_i and y'_i for similar constraints on the second tree.

Finally, the optimization criterion is

$$\min \sum_{i\in[n]} |x_i-x'_i| + \sum_{(i,j)\in switch} |x_i-x_j|$$

.

The first term is the sum of the distances of the match edges and the second term is the sum of the distances of the switch edges. As before, we will convert the absolute values to linear forms using standard techniques [3].

4 Experiments

In Clayton et al. [**?**] Figure 12.2B has 13 crossings. The integer program shows a drawing with only 10 crossings in the optimized drawing.

For Fig 10.6 in the book, the number of crossings is improved from 12 to 8 crossings.

5 Conclusions

In this paper we considered recent evolutionary events like host-switching, cohesion, and duplication by which one host (or parasite) species can be related to multiple parasite (or host) species. These are represented by tanglegrams with multiple edges between related leaves of the two trees. This generalizes the tanglegrams considered in the computer science literature. We showed that the XOR formulation for crossing mimization extends to the generalized tanglegrams. We also described integer programs for distance minimization that extend the integer programs from the earlier work. These show that the drawings considered in ecology literature can be improved to a drawing with fewer crossings.

References

[1] Balaji Venkatachalam, Jim Apple, Katherine St. John, Dan Gusfield, "Untangling Tanglegrams: Comparing Trees By Their Drawings". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 7, No. 4, Pages 588–597 2010.

[2] K. Buchin, M. Buchin, J. Byrka, M. Nollenburg, Y. Okamoto, R. I. Silveira, and A. Wolff. "Drawing (complete) binary tanglegrams: Hardness, approximation, fixed-parameter tractability". In *Graph Drawing*. Springer-Verlag, 2008.

[3] Dale Clayton, Kevin P. Johnson, and Sarah E. Bush. "Coevolution of Life on Hosts: Integrating Ecology and History", *Chicago University Press*, 2015.

[4] R. D. M. P. (Ed.). "Tangled Trees: Phylogeny, Cospeciation, and Coevolution". *University Of Chicago Press*, 2002.