Characterizing Planar Tanglegram Layouts and Applications to Edge Insertion Problems

Kevin Liu*

Department of Mathematics
University of Washington
Seattle, WA, U.S.A.

kliu15@uw.edu

Submitted: Jun 2, 2022; Accepted: May 8, 2023; Published: Jun 2, 2023

Abstract

Tanglegrams are formed by taking two rooted binary trees T and S with the same number of leaves and uniquely matching each leaf in T with a leaf in S. They are usually represented using layouts that embed the trees and matching in the plane. Given the numerous ways to construct a layout, one problem of interest is the Tanglegram Layout Problem, which is to efficiently find a layout that minimizes the number of crossings. This parallels a similar problem involving drawings of graphs, where a common approach is to insert edges into a planar subgraph. In this paper, we explore inserting edges into a planar tanglegram. Previous results on planar tanglegrams include a Kuratowski Theorem, enumeration, and an algorithm for finding a planar layout. We build on these results and characterize all planar layouts of a planar tanglegram. We then apply this characterization to construct a quadratic-time algorithm that inserts a single edge optimally. Finally, we generalize some results to multiple edge insertion.

Mathematics Subject Classifications: 05C05, 05C10, 05C30

1 Introduction

Let T and S be two rooted binary trees with leaves respectively labeled as $\{t_i\}_{i\in I}$ and $\{s_j\}_{j\in J}$, where $I, J \subseteq \mathbb{N}$ are finite index sets of the same size. If we let $\phi : I \to J$ be a bijection, then we can denote a tanglegram as (T, S, ϕ), where ϕ indicates that t_i is matched with $s_{\phi(i)}$. A *layout* of a tanglegram draws T, S, and the edges $(t_i, s_{\phi(i)})$ in the plane such that T is planarly embedded left of the line $x = 0$ with all leaves on $x = 0$, S is planarly embedded right of the line $x = 1$ with all leaves on $x = 1$, and the edges

*Supported by NSF grant DMS-1764012.

$(t_i, s_{\phi(i)})$ are drawn using straight lines. See Figure 1 for examples. A *crossing* is any pair of edges $(t_i, s_{\phi(i)})$ and $(t_j, s_{\phi(j)})$ that intersect in the layout, and the *crossing number* of a tanglegram (T, S, ϕ), denoted $\mathrm{crt}(T, S, \phi)$, is the minimum number of crossings over all layouts of (T, S, ϕ). The Tanglegram Layout Problem attempts to efficiently find a layout that achieves the crossing number.

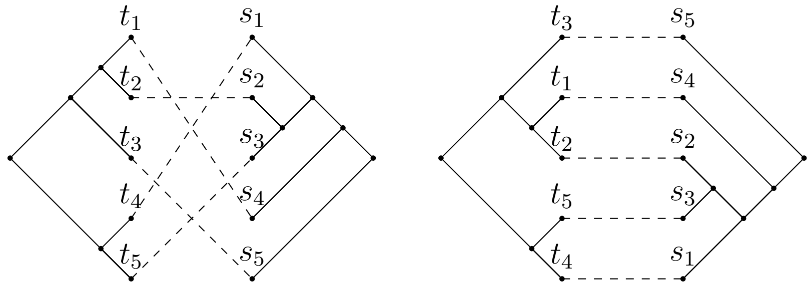

Figure 1: Two layouts for the same tanglegram, one with six crossings and one with no crossings.

Tanglegrams initially arose in biology and computer science. Biologists use binary trees to model evolution and tanglegrams to model relationships between species. Finding optimal layouts helps determine how two species may have co-evolved [16]. Applications in computer science include clustering, decomposition of programs into layers, or analyzing the difference in hierarchy between similar programs or different versions of the same program [3]. Combinatorial interest in tanglegrams developed more recently. Matsen et al. formalized tanglegrams as mathematical objects and described connections with phylogenetics [16]. Billey, Konvalinka, and Matsen then enumerated tanglegrams and constructed an algorithm to generate them uniformly at random [2]. Subsequently, Konvalinka and Wagner studied the properties of random tanglegrams [13], Ralaivaosaona, Ravelomanana, and Wagner counted planar tanglegrams [17], and Gessel counted several variations of tanglegrams using combinatorial species [8].

The crossing number of a tanglegram has connections with the crossing number of a graph G, denoted $\mathrm{cr}(G)$, which is the minimum number of crossings over all drawings of G. Determining if $\mathrm{cr}(G) \leqslant k$ for $k \in \mathbb{N}$ is NP-complete [7], and the same is true for determining if $\mathrm{crt}(T, S, \phi) \leqslant k$ [6]. Some of the known results in graph drawing have analogous results in tanglegram layouts, and some have approached the Tanglegram Layout Problem by translating what we know about graphs to tanglegrams. Czabarka, Székely, and Wagner recently used the well-known Kuratowski Theorem characterizing planarity of graphs to construct a Tanglegram Kuratowski Theorem characterizing *planar tanglegrams*, which are tanglegrams with crossing number zero [5]. Prior to this, Lozano et al. constructed their `Untangle` Algorithm for drawing a planar layout of a planar tanglegram [15]. Anderson et al. recently proved that removing a between-tree edge $(t_i, s_{\phi(i)})$ from a tanglegram reduces the crossing number by at most $n-3$, and they produced a family of tanglegrams to show that this bound is sharp [1]. They also found that the maximum crossing number over all tanglegrams asymptotically approaches $\frac{1}{2}\binom{n}{2}$, where n is the number of leaves in each tree.

Given the difficulty of minimizing crossings in graph drawings, some have studied approximating the minimum number of crossings rather than finding it exactly. One approach to this is edge insertion. The Edge Insertion Problem for graphs starts with a graph G and an edge $e \in G$ such that $G \setminus \{e\}$ is planar, and attempts to find an embedding of G into the plane so that the drawing of $G \setminus \{e\}$ is planar and the number of crossings from e is minimized. This problem is well studied. A linear-time algorithm exists to solve it, and some bounds have been found relating an optimal drawing of G and a solution to the Edge Insertion Problem for $G \setminus \{e\}$ with $\{e\}$ inserted [9, 10]. The Edge Insertion Problem generalizes to the Multiple Edge Insertion Problem, where we insert several edges $\{e_1, \ldots, e_n\}$ into planar $G \setminus \{e_1, \ldots, e_n\}$ optimally, and current approximation algorithms for graph drawing still use multiple edge insertion with planar subgraphs [4]. Given the role that edge insertion with planar subgraphs plays in graph drawings, it is plausible that edge insertion can play a similar role for tanglegram layouts. In this paper, we consider the tanglegram versions of the insertion problems for graphs, which we now state.

Problem (**Tanglegram Single Edge Insertion**). Given a tanglegram (T, S, ϕ) and a planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I = [n] \setminus \{i\}$ for $i \in [n]$, find a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ and has the minimal number of crossings possible.

Problem (**Tanglegram Multiple Edge Insertion**). Given a tanglegram (T, S, ϕ) and a planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I \subseteq [n]$, find a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ and has the minimal number of crossings possible.

We will start by characterizing the planar layouts of a planar tanglegram. For a planar tanglegram (T, S, ϕ), we will define a *leaf-matched pair* (u, v) as a pair of internal vertices $u \in T$ and $v \in S$ whose descendant leaves are matched by ϕ, and we will define an operation called a *paired flip*. These pairs of internal vertices have the property that their descendant leaves are matched by ϕ. By adding steps to the `Untangle` algorithm by Lozano et al. for drawing a planar layout of a planar tanglegram, we construct `ModifiedUntangle` (Algorithm 1), which also identifies leaf-matched pairs and stores them in a set L, obtaining the following result.

Theorem 1. *Let (T, S, ϕ) be a planar tanglegram, and let $\mathscr{P}(T, S, \phi)$ denote its collection of planar layouts. Let the output of `ModifiedUntangle`(T, S, ϕ) be the layout (X, Y) and set of leaf-matched pairs L. Every $(X', Y') \in \mathscr{P}(T, S, \phi)$ can be obtained by starting with (X, Y) and performing a sequence of paired flips at the leaf-matched pairs in L.*

Letting $\operatorname{size}(T, S, \phi)$ be the number of leaves in T or S, we then consider the generating function

$$F(x, q) = \sum_{\text{planar } (T,S,\phi)} x^{\operatorname{size}(T,S,\phi)} q^{|\{\text{leaf-matched pairs of } (T,S,\phi)\}|}. \tag{1}$$

The coefficient of $x^n q^k$ is the number of tanglegrams of size n with k leaf-matched pairs. When $\operatorname{size}(T, S, \phi) \geqslant 2$, the roots of T and S always form a leaf matched pair. A tanglegram is *irreducible* if this is the only leaf-matched pair. We let

$$H(x) = \sum_{\text{irreducible planar } (T,S,\phi)} x^{\text{size}(T,S,\phi)}, \tag{2}$$

where we use the convention that the coefficient of x^2 is $\frac{1}{2}$, as in [17, Proposition 8]. These generating functions have a relationship, which we can use to find coefficients of $x^n q^k$ in $F(x, q)$.

Theorem 2. *The generating function $F(x, q)$ satisfies the relation*

$$F(x, q) = x + q \cdot H(F(x, q)) + \frac{q \cdot F(x^2, q^2)}{2}. \tag{3}$$

Afterwards, we use our characterization of planar layouts to solve the Tanglegram Single Edge Insertion Problem. After considering various cases, we construct `Insertion` (Algorithm 5) and show the following result.

Theorem 3. *The `Insertion` Algorithm solves the Tanglegram Single Edge Insertion Problem in $O(n^2)$ time and space, where n is the size of the tanglegram.*

Finally, we consider Tanglegram Multiple Edge Insertion. Similar to the corresponding graph theory problems, Single Edge Insertion can be solved efficiently, but Multiple Edge Insertion is significantly more difficult.

Theorem 4. *The Tanglegram Multiple Edge Insertion Problem is NP-hard.*

Nevertheless, we generalize some of our results from `Insertion` to construct our `MultiInsertion` algorithm. While the space required will still be $O(n^2)$, the runtime is potentially exponential, depending on where vertices and edges are inserted, as well as how many leaf-matched pairs there are. Nevertheless, in certain situations, `MultiInsertion` efficiently solves the Tanglegram Multiple Edge Insertion Problem.

We start in Section 2 by outlining terminology, notation, and previous results. In Section 3, we give our `ModifiedUntangle` Algorithm, establish Theorem 1, and then show the relation in Theorem 2. In Section 4, we give our `Insertion` Algorithm and prove Theorem 1.3. In Section 5, we prove Theorem 1.4 and generalize some of our results from Section 4 to the Tanglegram Multiple Edge Insertion Problem. We conclude by discussing future work in Section 6.

2 Preliminaries

In this section, we begin by defining rooted binary trees and outlining the terminology and notation that we will use. We then introduce our notation for tanglegram layouts as ordered lists of leaves in the two trees. Afterwards, we define subtanglegrams, which correspond to subgraphs in graph theory. Note that removing vertices or edges in a tanglegram does not produce another tanglegram, so we give a description of the steps needed to construct subtanglegrams. An example of our notation and terminology will be given in Example 8, and the reader is encouraged to refer to this example as they read this section. We conclude with some known results.

A *rooted binary tree* T is a tree in which every vertex has either zero or two children, and where a designated vertex called the root, denoted $\mathrm{root}(T)$, is allowed to have degree two. A vertex that has children is called an *internal vertex*, and a vertex with no children is called a *leaf*. If v has children v_1 and v_2, we call v the parent of v_1 and v_2. We say vertex v_1 is a *descendant* of v_k or v_k is an *ancestor* of v_1 if there is a sequence of vertices $v_1, v_2, \dots, v_k$ such that v_{i+1} is the parent of v_i for $i = 1, 2, \dots, k-1$, and we use the notation $v_1 < v_k$ or $v_k > v_1$ to denote this. When needed, we use a subscript with the name of a tree to specify ancestry in that tree, such as $v_k >_T v_1$.

For an internal vertex $v \in T$, the *subtree rooted at* v is the tree formed by all vertices u with $u \leqslant v$, and this subtree then has v as its root. Using subtrees, we can represent trees using the nested lists notation from Section 2.3.2 of [12], where each set of parentheses represents a subtree. Unless otherwise stated, we will index leaves with $[n] = \{1, 2, \dots, n\}$, and usually we will omit labels for internal vertices.

All trees are considered up to isomorphism, so in particular, relabeling vertices does not produce a different tree. Given an internal vertex $v \in T$, a *flip* at vertex v is the operation that interchanges the order of the children for all $u \leqslant v$. Pictorially, if we start with a drawing of a tree T, a flip at $v \in T$ reflects the subtree rooted at v, which motivates the name "flip." Notice that each flip has order two, all flips commute with one another, and for any rooted binary tree T, flips generate all trees isomorphic to T.

Tanglegrams (T, S, ϕ) are formed from a pair of rooted binary trees T, S and a bijection ϕ matching their leaves. The *size* of (T, S, ϕ) is the common number of leaves in T or S. We will call the edges in T and S *tree edges* and call the edges induced by ϕ *between-tree edges*. For any vertex $u \in T$, we use $\mathrm{Lf}(u)$ to denote the leaves $\ell \in T$ such that $\ell \leqslant_T u$, and similarly for $\mathrm{Lf}(v)$ when $v \in S$. As with trees, we consider tanglegrams up to isomorphism. Relabeling vertices or replacing T and S with isomorphic trees does not produce a new tanglegram, provided that ϕ is modified appropriately. See [16] for more details. Notice that for any tanglegram (T, S, ϕ), flips will generate all tanglegrams isomorphic to (T, S, ϕ), as they generate all isomorphisms of the underlying trees.

Our notation for tanglegram layouts builds on the notation used in [15]. In any layout, the number of crossings is completely determined by the order of the leaves in the two trees and the bijection ϕ matching these leaves, as the between-tree edges $(t_i, s_{\phi(i)})$ and $(t_j, s_{\phi(j)})$ intersect when t_i is embedded above t_j and $s_{\phi(i)}$ is embedded below $s_{\phi(j)}$. Since we are primarily interested in the number of crossings rather than specific coordinates of the plane embedding, we give the following definition.

Definition 5. Let (T, S, ϕ) be a tanglegram drawn in the plane with a given layout. The *leaf order* of the given layout is a pair of ordered lists (X, Y), where X and Y respectively list the leaves of T and S in order of appearance from top to bottom in the layout.

One can view the leaf order of a layout (X, Y) as an equivalence class of layouts, where two layouts are equivalent if they draw the leaves of T and S in the same order from top to bottom. To recover a layout from the ordered lists (X, Y), one can draw the leaves listed in X and Y from top to bottom respectively on $x = 0$ and $x = 1$, and then use the information from T, S, and ϕ to draw the trees and between-tree edges. Flips generate all trees isomorphic to T or S, so they can act on leaf orders (X, Y) to obtain all possible leaf orders, where a flip at an internal vertex u acts on (X, Y) by reversing the order of the elements in $\mathrm{Lf}(u)$ in the appropriate list X or Y. Throughout this paper, we abuse terminology and refer to this pair of lists (X, Y) also as a tanglegram layout.

We will often decompose X and Y into concatenated lists $(X_1X_2 \dots X_m, Y_1Y_2 \dots Y_n)$, where each X_i or Y_j is some ordered collection of consecutive leaves. We do not impose any restrictions on X_i and Y_j beyond the fact that they must contain consecutive leaves, but usually we will use X_i or Y_j to represent the leaves in $\mathrm{Lf}(u)$ or $\mathrm{Lf}(v)$ for some $u \in T$ or $v \in S$. If we start with the layout $(X_1 \dots X_i \dots X_m, Y)$ and the sublist X_i contains the elements in $\mathrm{Lf}(u)$, we will denote the layout after a flip at u as $(X_1 \dots \overline{X_i} \dots X_m, Y)$, where the bar indicates that the sublist X_i is reversed. Note that while this notation using lists is convenient, the reader should visualize (X, Y) as embeddings into the plane, and when possible, we will include drawings to help illustrate the arguments that we make using these ordered lists.

Finally, we will define induced subtrees and induced subtanglegrams using a similar definition as in [5]. Notice that layouts (X', Y') of subtanglegrams correspond to taking sub-lists in layouts (X, Y) of the original tanglegram.

Definition 6. Let T be a tree with leaves indexed by $[n]$. For any $I \subseteq [n]$, the *rooted binary subtree induced by* I, denoted T_I, is formed by taking the minimal subtree of T containing the leaves indexed by I and suppressing all internal vertices that have only one child.

Definition 7. Let (T, S, ϕ) be a tanglegram with the leaves of T and S indexed by $[n]$. For any $I \subseteq [n]$, the *subtanglegram induced by* I is the tanglegram $(T_I, S_{\phi(I)}, \phi|_I)$, that is, the tanglegram formed from the induced subtrees T_I and $S_{\phi(I)}$ with leaves matched using the restriction $\phi|_I$.

Example 8. Consider the tanglegram (T, S, ϕ) with

$$T = (((t_1, t_2), t_3), (t_4, t_5)) \qquad S = (((s_1, (s_2, s_3)), s_4), s_5)$$

i	1	2	3	4	5
$\phi(i)$	4	2	5	1	3

Writing the leaves of T and S in the order given results in the layout $(t_1t_2t_3t_4t_5, s_1s_2s_3s_4s_5)$. A flip at the root of $((s_1, (s_2, s_3)), s_4)$ results in the layout $(t_1t_2t_3t_4t_5, s_4s_3s_2s_1s_5)$. By taking sub-lists of (X', Y') based on the elements in $\{1, 2, 4, 5\}$ and $\phi(\{1, 2, 4, 5\}) = \{1, 2, 3, 4\}$, we obtain the layout $(t_1t_2t_4t_5, s_4s_3s_2s_1)$ for the subtanglegram $(T_{\{1,2,4,5\}}, S_{\phi(\{1,2,4,5\})}, \phi|_{\{1,2,4,5\}})$. Using T and S, we produce the drawings in Figure 2 corresponding to these ordered lists.

We conclude this section with known results that will be used in our work. These results were previously mentioned, and we include them below for ease of citation. Recall that $\mathrm{crt}(T, S, \phi)$ is the minimum number of crossings over all layouts of a tanglegram (T, S, ϕ).

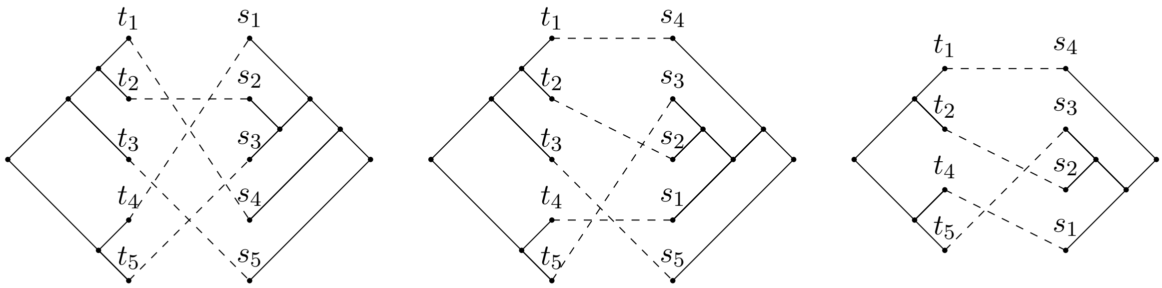

Figure 2: Layouts from Example 8.

Theorem 9. *[6, Theorem 3] The Tanglegram Layout Problem is NP-hard.*

Theorem 10. *[1, Theorem 3] Let (T, S, ϕ) be a tanglegram of size n and let $(T_I, S_{\phi(I)}, \phi|_I)$ be an induced subtanglegram of size $n-1$. Then $\operatorname{crt}(T, S, \phi) - \operatorname{crt}(T_I, S_{\phi(I)}, \phi|_I) \leqslant n - 3$.*

Theorem 11. *[1, Theorem 5] If (T, S, ϕ) is a tanglegram of size n, then $\operatorname{crt}(T, S, \phi) < \frac{1}{2}\binom{n}{2}$.*

3 Characterization of planar tanglegram layouts

In this section, we will start by giving our `ModifiedUntangle` algorithm, based on the `Untangle` algorithm by Lozano et al. in [15]. Our additional steps involve the set L. We will describe the significance of elements in this set, and then use it to characterize all planar layouts of a planar tanglegram. We conclude this section with our enumerative result on planar tanglegrams.

3.1 ModifiedUntangle Algorithm

The `Untangle` Algorithm by Lozano et al. starts by computing a table of Boolean values P, where $P[u, v]$ is True for $u \in T, v \in S$ if a descendant of u is matched with a descendant of v by ϕ. It begins with the ordered lists $X = (\operatorname{root}(T))$ and $Y = (\operatorname{root}(S))$ and then refines these lists by replacing vertices with their children in an order chosen based on the Boolean table P. They call these ordered lists of vertices (X, Y) from each step *partial layouts*, as they prescribe the order that some of the vertices in the tanglegram are drawn. `Untangle` terminates when the lists in (X, Y) contain only leaves of T and S, respectively. It then outputs this pair of lists (X, Y), which is an actual tanglegram layout. We give our modified version of `Untangle` below, which has additional steps involving a set L. Removing these steps yields the original `Untangle` algorithm. We will explain the significance of the set L in the next subsection.

Algorithm 1: `ModifiedUntangle` (based on [15, Algorithm 2])

Input: planar tanglegram (T, S, ϕ) with leaves $\{t_1, \dots, t_n\}$ and $\{s_1, \dots, s_n\}$
Output: a planar layout (X, Y) of (T, S, ϕ), list of leaf-matched pairs $L \subseteq T \times S$

$P :=$ Boolean table with $P[u, v] = False$ $\forall$ vertices $u \in T$, $v \in S$
set $P[t_i, s_{\phi(i)}] = True$ $\forall$ $i \in [n]$
recursively set $P[u, v] = True$ for internal vertices $u \in T, v \in S$ if there exists $u' \leqslant_T u$, $v' \leqslant_S v$ with $P[u', v'] = True$
$X := (\text{root}(T))$, $Y := (\text{root}(S))$ as ordered lists
$E := \{(\text{root}(T), \text{root}(S))\}$ as a set of edges
$L := \emptyset$
while *$X \cup Y$ contains an internal vertex of T or S* **do**
    $u :=$ internal vertex of $T \cup S$ with highest degree in the bipartite graph $G = (X, Y, E)$
    if $u \in X$ **then**
        if *u has degree 1 in G* **then**
            update $L := L \cup (u, v)$, where v is the unique neighbor of u in G
        update $X, E :=$ `Refine`(X, Y, u, E, P)
    else if $u \in Y$ **then**
        if *u has degree 1 in G* **then**
            update $L := L \cup (v, u)$, where v is the unique neighbor of u in G
        update $Y, E :=$ `Refine`(Y, X, u, E, P)
return $(X, Y), L$

Algorithm 2: `Refine` (based on [15, Algorithm 3])

Input: ordered lists of vertices (A, B), $u \in A$, edges E on $A \cup B$, Boolean table P
Output: A, E after u has been replaced with its children

$u_1, u_2 :=$ children of u in $T \cup S$
for $j \in [m]$ *such that* $(u, b_j) \in E$ *where* $B = (b_1, \dots, b_m)$ **do**
    update $E := E \setminus \{(u, b_j)\}$ `// delete edges involving` u
    for $i \in \{1, 2\}$ **do**
        if $P[u_i, b_j] = True$ **then**
            update $E := E \cup \{(u_i, b_j)\}$ `// insert edges involving` u_1 `or` u_2
$k := \max\{j \in [m] : (u_1, b_j) \in E\}$ `// last vertex in` B `adjacent to` u_1
if *$j > k$ for all $(u_2, b_j) \in E$* **then**
    replace u with $u_1 u_2$ in A
else
    replace u with $u_2 u_1$ in A
return A, E

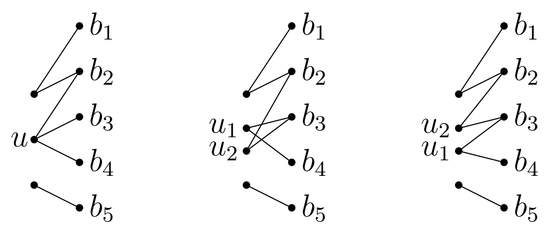

Figure 3: A visualization of the `Refine` *algorithm. Draw the inputs as a bipartite graph $G = (A, E, B)$ shown on the left, where the vertices are drawn from top to bottom based on the order of elements in A and B. To refine the vertex u of highest degree, consider the embedding shown in the middle with u_1 drawn above u_2, where new edges involving u_1 and u_2 are drawn using the Boolean table P. Since this drawing has crossings,* `Refine` *will replace u with u_2u_1. Notice that in this case, drawing u_2 above u_1 results in a planar drawing, as shown on the right.*

A description of the steps in `Refine` is shown in Figure 3. Note that `Refine` does not actually check if drawing u_2 above u_1 results in a planar embedding. This is because planarity of (T, S, ϕ) guarantees that at least one of these embeddings will be planar, provided we made appropriate choices at previous refinements. Details are addressed in the proof of the next theorem. Since results from this proof will be relevant for our work, we repeat it below for the convenience of the reader. Note that Lemma 13 is stated more generally than in [15]. This extra generality will be useful in the next subsection.

Definition 12. A partial layout (X, Y) for a tanglegram is called *promising* if it can be extended to a planar layout by successively replacing vertices with their children in some order.

Lemma 13 ([15], Lemma 3)**.** *Let (X, Y) be a promising partial layout of a planar tanglegram (T, S, ϕ), and let E be the set of edges on $X \cup Y$ generated using the Boolean table P, that is, for all $u \in X$ and $v \in Y$, $(u, v) \in E$ if and only if $P[u, v] = True$. Let u be a vertex of highest degree in the bipartite graph (X, E, Y).*

(a) If $\deg(u) = 1$, then replacing u with u_1u_2 or u_2u_1 results in a promising partial layout.

(b) If $\deg(u) > 1$, then either replacing u with u_1u_2 or replacing u with u_2u_1 results in a promising partial layout, but not both.

In particular, if (X, Y) is promising at the beginning of an iteration of the ***while*** *loop in* `Untangle`*, then it is promising at the end.*

Proof. Without loss of generality, we will assume $u \in T$, as the result when $u \in S$ is done similarly. We let (X_1, Y) and (X_2, Y) be the partial layouts obtained by replacing u with u_1u_2 or u_2u_1, respectively. Since (X, Y) is promising, at least one of these partial layouts must be promising, so assume that (X_1, Y) is promising.

First, suppose $\deg(u) = 1$ in (X, E, Y). Since u is a vertex of maximum degree, the unique neighbor of u, denoted v, also has degree 1. Notice that since u and v have degree 1, $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$ must be matched by ϕ. Extend (X_1, Y) to a planar layout (X', Y') of (T, S, ϕ), where $\mathrm{Lf}(u_1)$ appears before $\mathrm{Lf}(u_2)$. If we perform a flip at u and a flip at v, then we obtain a layout where $\mathrm{Lf}(u_2)$ appears before $\mathrm{Lf}(u_1)$, as shown in Figure 4. Notice that this is a planar layout that can be obtained by replacing u with u_2u_1 instead, implying (X_2, Y) is also promising. We see that regardless of how we replace u with u_1 and u_2, the resulting partial layout is promising.

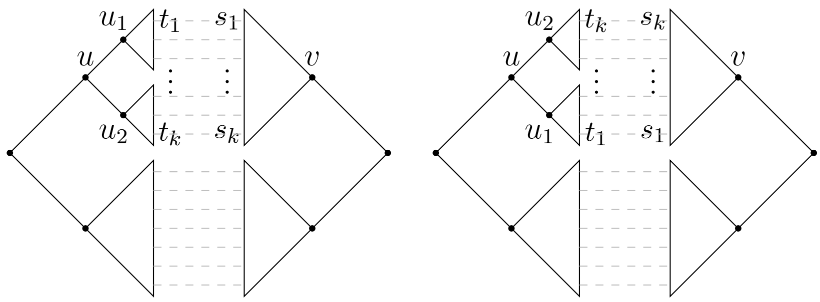

Figure 4: Starting with the layout (X', Y') on the left with $\{t_i\}_{i=1}^k$ and $\{s_i\}_{i=1}^k$ denoting leaves, performing flips at u and v produces another planar layout.

Next, suppose that $\deg(u) > 1$. As before, we suppose (X_1, Y) is promising. Let E' be the edges on $X_1 \cup Y$ constructed using the Boolean table P, and for $i = 1, 2$, we let $N(u_i)$ denote the set of neighbors of u_i in (X_1, E', Y). We claim that $N(u_1)\Delta N(u_2) \neq \emptyset$, where Δ denotes the symmetric difference of two sets. To see this, suppose that $N(u_1) = N(u_2)$. If $|N(u_1)| = |N(u_2)| = 1$, then this would imply $\deg(u) = 1$ in (X, E, Y), which by assumption cannot be the case. Hence, $|N(u_1)| = |N(u_2)| \geqslant 2$. Then there exists some pair of vertices $v_1, v_2 \in N(u_1) = N(u_2)$ that are each adjacent to both u_1 and u_2. However, this implies a crossing occurs in both (X_1, E', Y) and (X_2, E', Y), as shown in Figure 5. Then by construction of the Boolean table P, there exist some $t_{1,1}, t_{1,2} \in \mathrm{Lf}(u_1)$ and $t_{2,1}, t_{2,2} \in \mathrm{Lf}(u_2)$ where each $t_{i,j} \in \mathrm{Lf}(u_i)$ is matched to some $s_{i,j} \in \mathrm{Lf}(v_j)$. Regardless of any refinements of (X, Y), the resulting layout will have either the crossing $(t_{1,2}, s_{1,2})$ and $(t_{2,1}, s_{2,1})$, or the crossing $(t_{1,1}, s_{1,1})$ and $(t_{2,2}, s_{2,2})$. Since (X_1, Y) is assumed to be promising, it must be that $N(u_1) \neq N(u_2)$.

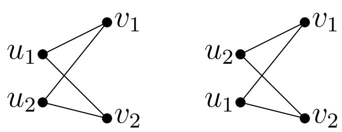

Figure 5: Arrangements of the vertices u_1, u_2, v_1 and v_2 when $N(u_1) = N(u_2)$ and $\deg(u) \geqslant 2$.

With $N(u_1)\Delta N(u_2) \neq \emptyset$ established, we let $v \in N(u_1)\Delta N(u_2)$. First, we assume that $v = v_1 \in N(u_1) \setminus N(u_2)$. Since (X_1, Y) is promising, it must be that drawing (X_1, E', Y)

with vertices in the orders indicated by X_1 and Y produces no crossings. Then v_1 must appear before any $v_2 \in N(u_2)$ in the list Y. Furthermore, for any $v_2 \in N(u_2)$, if we interchange u_1 and u_2, then the edges (u_1, v_1) and (u_2, v_2) will intersect. Regardless of our future refinements, there will exist some leaves $t_i \in \text{Lf}(u_1), t_j \in \text{Lf}(u_2), s_{\phi(i)} \in \text{Lf}(v_1), s_{\phi(j)} \in \text{Lf}(v_2)$ such that $(t_i, s_{\phi(i)})$ and $(t_j, s_{\phi(j)})$ intersect, which implies (X_2, Y) cannot be promising. A similar argument applies when $v = v_2 \in N(u_2) \setminus N(u_1)$.

Now consider the **while** loop in `Untangle`. The algorithm uses `Refine` on a vertex u of highest degree in (X, E, Y), which replaces u with u_1 and u_2. If $\deg(u) = 1$, then (a) shows that (X, Y) is promising regardless of the choice at u. If $\deg(u) > 1$, `Refine` replaces (X, Y) with (X_1, Y) or (X_2, Y) based on whichever bipartite graph (X_1, E', Y) or (X_2, E', Y) does not have crossings, and our proof of (b) shows that this results in a promising partial layout. □

Theorem 14. *[15, Theorem 1] For any planar tanglegram (T, S, ϕ), the* `Untangle` *algorithm terminates in a planar layout (X, Y).*

Proof. If (T, S, ϕ) is planar, then $(X, Y) = (\text{root}(T), \text{root}(S))$ in line 4 is promising. By Lemma 13, (X, Y) is promising after each iteration of the **while** loop. This loop terminates when (X, Y) contains only leaves of T and S, which must then be a planar layout. □

Remark 15. In the proofs of Lemma 13 and Theorem 14, the arguments hold regardless of which vertex of highest degree u is selected. In fact, we do not even need to select the vertex of highest degree at every step in `Untangle`. We specified a vertex of highest degree for simplicity. As long as `Untangle` does not use `Refine` on a vertex in (X, E, Y) with degree one while its neighbor has degree more than one, the algorithm will still output a planar layout for a planar tanglegram.

Lemma 16. *[15, Lemma 4]* `Untangle` *runs in $O(n^2)$ time and space, where n is the size of the tanglegram.*

Remark 17. The additional steps in `ModifiedUntangle` involve the set L, which has size at most $n - 1$. Hence, for a planar tanglegram of size n, Theorem 14 and Lemma 16 also imply that `ModifiedUntangle` terminates in a planar layout (X, Y) and runs in $O(n^2)$ time and space.

3.2 Leaf-matched pairs and paired flips

We now consider the additional steps involving the set L in `ModifiedUntangle` algorithm. While the `Untangle` algorithm produces a planar layout for planar tanglegrams, one might ask how to generate all of them. As noted in the proof of Lemma 13, if (X, Y) is a planar layout of (T, S, ϕ), then one method to generate additional planar layouts is using flips at vertices $u \in T$ and $v \in S$ where $\text{Lf}(u)$ and $\text{Lf}(v)$ are matched by ϕ. We give a name for these pairs of vertices and the operation involving a flip at both vertices, followed by an example in Figure 6. Then we show that `ModifiedUntangle` identifies these pairs in the set L.

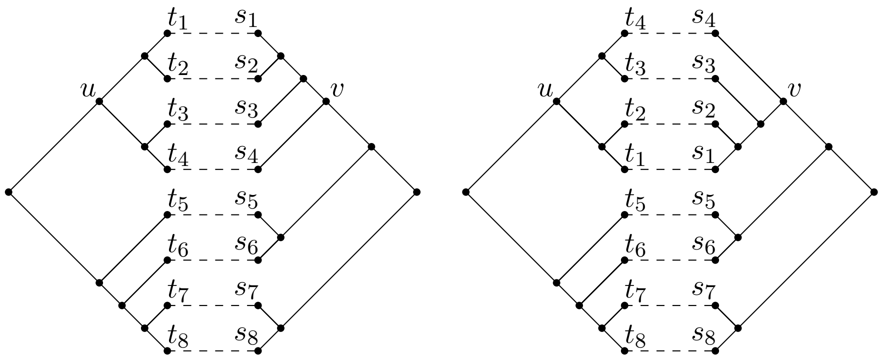

Figure 6: A paired flip at (u, v) maps each layout to the other one.

Definition 18. Let (T, S, ϕ) be a tanglegram with layout (X, Y). A pair of internal vertices (u, v) with $u \in T$ and $v \in S$ is a *leaf-matched pair* of (T, S, ϕ) if $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$ are matched by ϕ. A *paired flip* at (u, v) is the operation on (T, S, ϕ) corresponding to a flip at u and a flip at v. This maps (X, Y) to the layout (X', Y'), where X' is the image of X after a flip at u and Y' is the image of Y after a flip at v.

Lemma 19. *Let (T, S, ϕ) be a planar tanglegram. A pair of internal vertices (u, v) is a leaf-matched pair of a planar tanglegram (T, S, ϕ) if and only if at some step of the `ModifiedUntangle` algorithm, the internal vertices $u \in T$ and $v \in S$ appear as adjacent degree one vertices in (X, E, Y). Hence, the set L returned by `ModifiedUntangle` is the set of leaf-matched pairs of (T, S, ϕ).*

Proof. Suppose the internal vertices $u \in T$ and $v \in S$ appear as adjacent degree one vertices in (X, E, Y) during some step of the `ModifiedUntangle` algorithm. Since both vertices have degree one, the construction of the edges in (X, E, Y) using the Boolean table P in line 4 of `Refine` implies that $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$ are matched under ϕ, and thus (u, v) is a leaf-matched pair.

Conversely, suppose (u, v) is a leaf-matched pair. If u and v are the root vertices of T and S, then these appear as degree one vertices at the first step of `ModifiedUntangle`. Otherwise, at some step of the algorithm, either u or v will appear for the first time in a partial layout (X, Y), and without loss of generality, we assume it is $u \in X$. Since v has not appeared in a partial layout yet, there is some vertex $v' \in Y$ that is an ancestor of v. Then $\mathrm{Lf}(u)$ is matched with a proper subset of $\mathrm{Lf}(v')$ in the tanglegram (T, S, ϕ), so $\deg(v') > 1$ in (X, E, Y). From line 8 of `ModifiedUntangle`, we see that v' will be replaced with its children before u is. Repeating this argument, v will eventually appear before we use `Refine` on u, and at this time, u and v will be adjacent degree one vertices since they are a leaf-matched pair of (T, S, ϕ). □

We know that given a planar layout (X, Y) of (T, S, ϕ), paired flips will generate additional planar layouts, but we do not yet know that they generate all possible planar layouts. It may be possible that some appropriate choice of flips not equivalent to a sequence of paired flips also results in a planar layout. We will show that this is in fact not the case. Our proof for this uses Lemma 13, which holds for all promising partial layouts, not just those considered in `ModifiedUntangle`.

First, notice that for a tanglegram (T, S, ϕ) of size n, `ModifiedUntangle` starts with a promising partial layout $(X_1, Y_1) = (\mathrm{root}(T), \mathrm{root}(S))$. At each step, it replaces an internal vertex of T or S using `Refine`. Since a tree on n leaves has $n-1$ internal vertices, the algorithm uses `Refine` a total of $2n-2$ times. Thus, it produces a sequence of promising partial layouts $\{(X_k, Y_k)\}_{k=1}^{2n-1}$ that terminates at $(X, Y) = (X_{2n-1}, Y_{2n-1})$. We give a name for such a sequence.

Definition 20. Let (T, S, ϕ) be a tanglegram with layout (X, Y). We call $\{(X_k, Y_k)\}_{k=1}^{2n-1}$ a *partial sequence for* (X, Y) if

- for $k = 1, 2, \ldots, 2n-1$, (X_k, Y_k) is a partial layout,
- $(X_1, Y_1) = (\mathrm{root}(T), \mathrm{root}(S))$,
- $(X_{2n-1}, Y_{2n-1}) = (X, Y)$, and
- for $k = 1, 2, \ldots, 2n-2$, the partial layout (X_{k+1}, Y_{k+1}) is obtained from (X_k, Y_k) by refining some vertex $u \in X_k \cup Y_k$, that is, replacing u with its children in an appropriate order.

A partial sequence is *promising* if all (X_k, Y_k) are promising partial layouts, or equivalently, if $(X, Y) = (X_{2n-1}, Y_{2n-1})$ is a planar layout.

If (X', Y') is another planar layout of (T, S, ϕ), we can use $\{(X_k, Y_k)\}_{k=1}^{2n-1}$ to find a promising partial sequence $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ for (X', Y'). We do this by constructing each (X'_k, Y'_k) as follows.

- Draw the trees T and S with leaves from top to bottom in the order described by (X', Y').
- For each $u \in X_k$, contract the subtree of T rooted at u to the vertex u itself. Do the same for each $v \in Y_k$, and call the resulting trees T_k and S_k.
- Let X'_k be the leaves of T_k listed from top to bottom, and similarly for Y'_k and S_k.

An example of these steps is shown below in Figure 7. Note that by construction, (X_k, Y_k) and (X'_k, Y'_k) contain the same vertices, but possibly in different orders. We now show that $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ constructed in this manner is a promising partial sequence for (X', Y') and then use this to establish Theorem 1.

Lemma 21. *The sequence* $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ *is a promising partial sequence for* (X', Y').

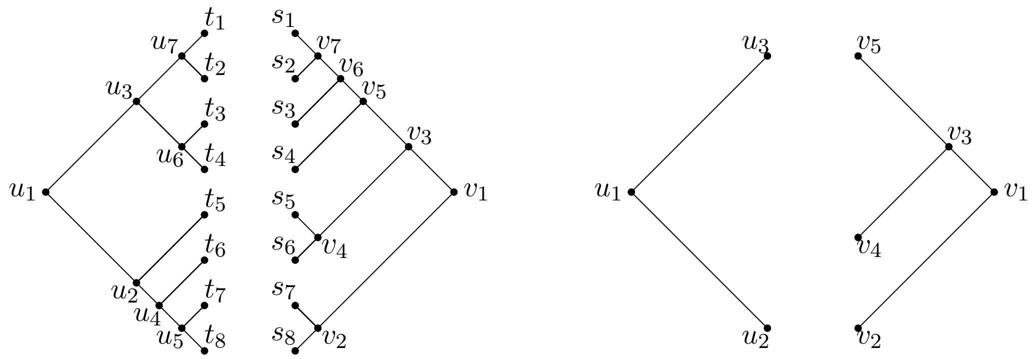

Figure 7: Starting with drawings of T and S corresponding to (X', Y'), we use $(X_k, Y_k) = (u_2u_3, v_2v_4v_5)$ to form the contracted trees T_k and S_k shown on the right. Listing leaves from top to bottom gives us the partial layout $(X'_k, Y'_k) = (u_3u_2, v_5v_4v_2)$.

Proof. By construction, each (X'_k, Y'_k) is a partial layout, $(X'_1, Y'_1) = (\mathrm{root}(T), \mathrm{root}(S))$, and $(X'_{2n-1}, Y'_{2n-1}) = (X', Y')$. It remains to show that refining a vertex $u \in X'_k \cup Y'_k$ produces (X'_{k+1}, Y'_{k+1}). Denote the vertex refined in (X_k, Y_k) to obtain (X_{k+1}, Y_{k+1}) as u_k, and without loss of generality, we assume $u_k \in X_k$. This implies that (X'_k, Y'_k) and (X'_{k+1}, Y'_{k+1}) have almost the same vertices, except (X'_k, Y'_k) contains u_k, while (X'_{k+1}, Y'_{k+1}) contains its two children $u_{k,1}$ and $u_{k,2}$. By our construction using contractions, the tree T_k can be obtained from T_{k+1} by contracting the two children of u_k onto the vertex itself. Thus, we can obtain (X'_k, Y'_k) from (X'_{k+1}, Y'_{k+1}) by replacing the adjacent children of u_k with u_k itself. Then we can also obtain (X'_{k+1}, Y'_{k+1}) from (X'_k, Y'_k) by refining u_k. Thus, all conditions in Definition 20 are satisfied, so we see that $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ is a promising partial sequence for (X', Y'). □

Proof of Theorem 1. By Lemma 19, L is the set of leaf-matched pairs of (T, S, ϕ). It is clear that if $(u, v) \in L$, then starting with (X, Y) and performing a paired flip at (u, v) produces another layout in $\mathscr{P}(T, S, \phi)$, so the same is true if we perform any sequence of paired flips starting with (X, Y). We show that all planar layouts can be obtained this way.

Let $(X', Y') \in \mathscr{P}(T, S, \phi)$ be distinct from (X, Y). Let $\{(X_k, Y_k)\}_{k=1}^{2n-1}$ be the promising partial sequence for (X, Y) produced in `ModifiedUntangle`, with corresponding bipartite graphs $\{(X_k, E_k, Y_k)\}_{k=1}^{2n-1}$. By Lemma 21, we can use this sequence for (X, Y) to construct a corresponding promising partial sequence $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ for (X', Y') where (X_k, Y_k) and (X'_k, Y'_k) contain the same vertices, though possibly in different orders. Thus, if we use the Boolean table P to construct edges E'_k on the vertices in $X'_k \cup Y'_k$, then $E'_k = E_k$. Since $(X', Y') \neq (X, Y)$, there must be some minimal m such that $(X_{m+1}, Y_{m+1}) \neq (X'_{m+1}, Y'_{m+1})$. Without loss of generality, we assume the refined vertex at this step was $u \in X_m$ and that `Refine` replaced u with u_1u_2 to obtain (X_{m+1}, Y_{m+1}), while (X'_{m+1}, Y'_{m+1}) requires replacing u with u_2u_1.

Consider the degree of u in the bipartite graph $(X_m, E_m, Y_m) = (X'_m, E_m, Y'_m)$. If $\deg(u) \geqslant 2$, then Lemma 13 implies that (X'_{m+1}, Y'_{m+1}) is not promising, which is not the case since $\{(X'_k, Y'_k)\}_{k=1}^{2n-1}$ is a promising partial sequence for (X', Y'). Thus, it must be that $\deg(u) = 1$. Since `ModifiedUntangle` always refines a vertex of highest degree, this implies that all vertices in $(X_m, E_m, Y_m) = (X'_m, E_m, Y'_m)$ must have degree 1.

Let $v \in Y_m$ be the unique neighbor of u in (X_m, E_m, Y_m). Once we replace u with its children, notice that v will be the unique vertex in $(X_{m+1}, E_{m+1}, Y_{m+1})$ with $\deg(v) \geqslant 2$. Thus, after `ModifiedUntangle` replaces u with u_1u_2 to obtain (X_{m+1}, Y_{m+1}), it will replace v with its children in some order to obtain (X_{m+2}, Y_{m+2}).

Lemma 19 implies that $(u, v) \in L$, so we let $(\widetilde{X}, \widetilde{Y}) \in \mathscr{P}(T, S, \phi)$ be (X, Y) after a paired flip at (u, v). Using $\{(X_k, Y_k)\}_{k=1}^{2n-1}$, we again use Lemma 21 to construct a promising partial sequence $\{(\widetilde{X}_k, \widetilde{Y}_k)\}_{k=1}^{2n-1}$ for $(\widetilde{X}, \widetilde{Y})$. By construction, $(\widetilde{X}_k, \widetilde{Y}_k) = (X_k, Y_k)$ for all $k \leqslant m$, and because of the paired flip at (u, v), we see that $(\widetilde{X}_k, \widetilde{Y}_k) = (X'_k, Y'_k)$ for all $k \leqslant m + 1$. Furthermore, the preceding paragraph implies that $(\widetilde{X}_{m+2}, \widetilde{Y}_{m+2})$ is obtained from $(\widetilde{X}_{m+1}, \widetilde{Y}_{m+1})$ by refining the vertex v. Since $\deg(v) = 2$ in $(\widetilde{X}_{m+1}, E_{m+1}, \widetilde{Y}_{m+1})$, Lemma 13 implies that a unique choice for the children of v results in a promising partial layout, and thus it must be that $(\widetilde{X}_{m+2}, \widetilde{Y}_{m+2}) = (X'_{m+2}, Y'_{m+2})$.

If $(\widetilde{X}, \widetilde{Y}) \neq (X', Y')$, we can repeat the above argument. Eventually, this process terminates in a planar layout $(\widetilde{X}, \widetilde{Y})$ obtained from a sequence of paired flips starting at (X, Y), where $(\widetilde{X}_k, \widetilde{Y}_k) = (X'_k, Y'_k)$ for all k. Hence, we see that $(X', Y') = (\widetilde{X}, \widetilde{Y})$, and any $(X', Y') \in \mathscr{P}(T, S, \phi)$ can be obtained using a sequence of paired flips starting with (X, Y). □

Remark 22. A tanglegram is *irreducible* if its only leaf matched pair is given by the roots of the two trees. For irreducible tanglegrams (T, S, ϕ) of size at least three, Theorem 1 implies that there are exactly two planar layouts, which must be mirror images of one another. This specific case was established and used by Ralaivaosaona, Ravelomanana, and Wagner to enumerate planar tanglegrams [17, Proposition 5].

We now define an undirected graph called the flip graph of a planar tanglegram. Theorem 1 gives us a corollary about this graph. By using the outputted layout from `ModifiedUntangle` and considering all subsets of L, one could in principle find all planar layouts of a planar tanglegram (T, S, ϕ) and produce the flip graph of a tanglegram, though there may be exponentially many planar layouts.

Definition 23. Let (T, S, ϕ) be a planar tanglegram. Define the *flip graph of* (T, S, ϕ) as $\Gamma(T, S, \phi) = (V, E)$ with vertices $v_{(X,Y)} \in V$ corresponding to planar layouts (X, Y), and edges $(v_{(X,Y)}, v_{(X',Y')}) \in E$ if (X', Y') can be obtained from (X, Y) by a paired flip at some leaf-matched pair (u, v) of (T, S, ϕ).

Corollary 24. *The flip graph of a planar tanglegram is connected.*

3.3 Enumeration of planar tanglegrams by number of leaf-matched pairs

In this subsection, we show an enumerative result for the number of planar tanglegrams of size n with k leaf-matched pairs. Recall the generating functions $F(x, q)$ and $H(x)$

from (1) and (2). We first give a definition, and then establish our generalization of [17, Theorem 1].

Definition 25. The *irreducible component* of a tanglegram (T, S, ϕ), denoted $\mathrm{Irr}(T, S, \phi)$, is the irreducible tanglegram formed by contracting each non-root leaf-matched pair of (T, S, ϕ) to a single pair of matched leaves.

Proof of Theorem 2. Equation (3) is equivalent to

$$F(x,q) = x + q \cdot \left(H(F(x,q)) - \frac{F(x,q)^2}{2} \right) + q \cdot \frac{F(x,q)^2 + F(x^2,q^2)}{2}, \tag{4}$$

so we establish this relation instead. The term x accounts for the unique tanglegram of size 1, which has no leaf-matched pairs. For the remaining tanglegrams, we can form each tanglegram (T, S, ϕ) by starting with its irreducible component $\mathrm{Irr}(T, S, \phi)$ and replacing matched leaves with planar tanglegrams (possibly of size 1). We consider two cases depending on the size of $\mathrm{Irr}(T, S, \phi)$.

First, consider tanglegrams with $\mathrm{size}(\mathrm{Irr}(T, S, \phi)) \geqslant 3$. As noted in the proof of [17, Theorem 1], these tanglegrams do not have any symmetry. The generating function $q \cdot [H(x) - x^2/2]$ counts irreducible planar tanglegrams of size $n \geqslant 3$. A term qx^n corresponds to a planar irreducible tanglegram of size n, and replacing a pair of matched leaves with a planar tanglegram corresponds to replacing x with $F(x,q)$. Hence, $q \cdot [H(F(x,q)) - F(x,q)^2/2]$ enumerates tanglegrams with irreducible component of size $n \geqslant 3$.

Second, suppose $\mathrm{size}(\mathrm{Irr}(T, S, \phi)) = 2$. These tanglegrams are formed by starting with the unique planar tanglegram of size two corresponding to the term qx^2 and replacing the two pairs of leaves with two planar tanglegrams $\{(T_1, S_1, \phi_1), (T_2, S_2, \phi_2)\}$, where the order is not relevant. The generating function $F(x,q)^2$ would count ordered pairs of planar tanglegrams. This correctly counts the case when (T_1, S_1, ϕ_1) and (T_2, S_2, ϕ_2) are the same, but counts all other cases twice. To remedy this over-counting, we can add $F(x^2, q^2)$, which counts the pairs where (T_1, S_1, ϕ_1) and (T_2, S_2, ϕ_2) are the same, and then divide the result by two to account for the order not being relevant. Hence, $q \cdot \frac{F(x,q)^2 + F(x^2,q^2)}{2}$ enumerates tanglegrams with irreducible component of size two. Combined, we obtain (4). $\square$

n, k	1	2	3	4	5	6	total
2	1						1
3	1	1					2
4	5	4	2				11
5	34	28	11	3			76
6	273	239	102	29	6		649
7	2436	2283	1045	325	73	11	6173

Table 1: Coefficients of $x^n q^k$ in $F(x,q)$ for $2 \leqslant n \leqslant 7$.

Note that substituting $q = 1$ results in the original relation given in [17, Theorem 1]. Using this result and the coefficients of $H(x)$ from [17], it takes a computer-algebra

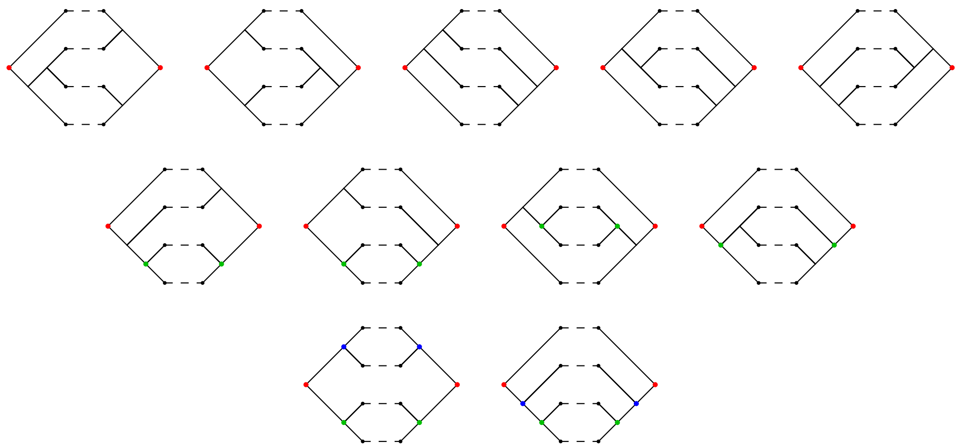

Figure 8: The 11 planar tanglegrams of size 4. The first five tanglegrams are irreducible, the next four have two leaf-matched pairs, and the final two have three pairs.

system only a few seconds to generate several coefficients of $F(x, q)$. We collect some of these coefficients in Table 1. See [11, A349409] for more terms. The corresponding planar tanglegrams for $n = 4$ are shown in Figure 8.

4 The Tanglegram Single Edge Insertion Problem

In this section, we solve the Tanglegram Single Edge Insertion Problem. For convenience of the reader, we restate the problem below.

Problem (**Tanglegram Single Edge Insertion**). Given a tanglegram (T, S, ϕ) and a planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I = [n] \setminus \{i\}$ for $i \in [n]$, find a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ and has the minimal number of crossings possible.

Since planar layouts of $(T_I, S_{\phi(I)}, \phi|_I)$ are relevant for this problem, we can apply our work from Section 3. We use the notation $(T, S, \phi|_I) = (T, S, \phi) \setminus \{(t_j, s_{\phi(j)})\}_{j \notin I}$ for a tanglegram with some between-tree edges removed. Note that $(T, S, \phi|_I)$ is well-defined as an input into `ModifiedUntangle`, motivating this notation. Letting $(X, Y), L =$ `ModifiedUntangle`$(T, S, \phi|_I)$, we will show that (X, Y) does restrict to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ whenever it is a planar subtanglegram. However, L may contain pairs (u, v) where $u \in T$ or $v \in S$ are not vertices in $(T_I, S_{\phi(I)}, \phi|_I)$. One simple example of this situation is shown in Figure 9.

To find the leaf-matched pairs of the subtanglegram, one could input $(T_I, S_{\phi(I)}, \phi|_I)$ itself into `ModifiedUntangle`, and then relate the outputs to (T, S, ϕ), though this would require extending the outputted layout of $(T_I, S_{\phi(I)}, \phi|_I)$ to a layout for (T, S, ϕ). Alternatively, we can directly construct the set of leaf-matched pairs of $(T_I, S_{\phi(I)}, \phi|_I)$ using L.

Definition 26. Let (T, S, ϕ) be a tanglegram of size n, and let $I \subseteq [n]$. For any pair of vertices $u \in T$, $v \in S$ such that $\mathrm{Lf}(u) \cap T_I$ is paired with $\mathrm{Lf}(v) \cap S_{\phi(I)}$ by $\phi|_I$, define their

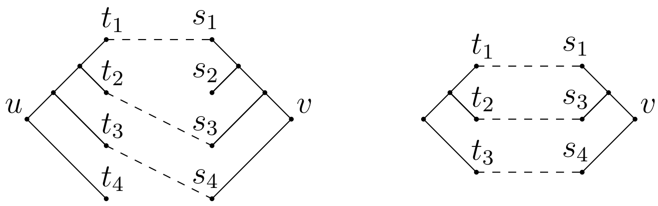

Figure 9: When used on $(T, S, \phi|_{\{1,2,3\}})$ shown on the left, `ModifiedUntangle` *will add (u, v) to L on the first step of the algorithm. However, u is not a vertex of the subtanglegram $(T_{\{1,2,3\}}, S_{\{1,3,4\}}, \phi|_{\{1,2,3\}})$, shown on the right.*

reduced vertices to be

$$u_{red} = \text{minimal } u' \leqslant_T u \text{ such that } \mathrm{Lf}(u) \cap T_I = \mathrm{Lf}(u') \cap T_I$$

$$v_{red} = \text{minimal } v' \leqslant_S v \text{ such that } \mathrm{Lf}(v) \cap S_{\phi(I)} = \mathrm{Lf}(v') \cap S_{\phi(I)}.$$

Given the set L from `ModifiedUntangle`$(T, S, \phi|_I)$, define $L(I)$ to be the set of all (u_{red}, v_{red}) such that $(u, v) \in L$ and $|\mathrm{Lf}(u) \cap T_I| = |\mathrm{Lf}(v) \cap S_{\phi(I)}| > 1$.

Lemma 27. *Suppose (T, S, ϕ) is a tanglegram with planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I \subseteq [n]$, and let $(X, Y), L =$ ModifiedUntangle$(T, S, \phi|_I)$. If $(u, v) \in L$, then $\mathrm{Lf}(u) \cap T_I$ and $\mathrm{Lf}(v) \cap S_{\phi(I)}$ are matched by $\phi|_I$.*

Proof. Since $(u, v) \in L$, we know u and v appeared as adjacent degree one vertices in a bipartite graph (X_j, E_j, Y_j) during some step of `ModifiedUntangle`. The edges E_j are constructed using the Boolean table P, and because of the restriction $\phi|_I$, $P[u, v] = True$ if and only if some element of $\mathrm{Lf}(u) \cap T_I$ is matched with an element of $\mathrm{Lf}(v) \cap S_{\phi(I)}$ by $\phi|_I$. From this, we see that u and v being adjacent degree one vertices in (X_j, E_j, Y_j) implies $\mathrm{Lf}(u) \cap T_I$ and $\mathrm{Lf}(v) \cap S_{\phi(I)}$ are matched by $\phi|_I$. □

Lemma 28. *Suppose (T, S, ϕ) is a tanglegram with planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I \subseteq [n]$, and let $(X, Y), L =$ ModifiedUntangle$(T, S, \phi|_I)$.*

(a) The layout (X, Y) restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$.

(b) The set $L(I)$ is the set of leaf-matched pairs of $(T_I, S_{\phi(I)}, \phi|_I)$.

Proof. To prove (a), we use a similar argument as in Theorem 14. Call (X, Y) promising if it extends to a planar drawing of $(T, S, \phi|_I)$. The initial partial layout of roots is promising by assumption. We claim that (X, Y) is still promising after an iteration of `Refine`. Suppose we are refining a vertex of highest degree $u \in X$. It is possible that when we use `Refine` on u to replace it with its children u_1 and u_2, the Boolean table values $P[u_i, v]$ are false for all $v \in Y$, and this occurs precisely when all between-tree edges incident to leaves in $\mathrm{Lf}(u_i)$ have been removed. If this occurs, we assume without loss of generality that it occurs for u_1. Since (X, Y) was promising, it extends to some planar drawing, and if we interchange the subtrees rooted at u_1 and u_2, we obtain another planar drawing since leaves in $\mathrm{Lf}(u_1)$ are not incident to any of the between-tree edges. Thus, replacing u with u_1u_2 or u_2u_1 both result in a promising partial layout. For all other cases, the same arguments from Lemma 13 apply, and thus `ModifiedUntangle` terminates in a drawing of $(T, S, \phi|_I)$ that is planar. Restricting to just the leaves in I, this must be a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$.

For (b), we let $(X', Y'), L' = \texttt{ModifiedUntangle}(T_I, S_{\phi(I)}, \phi|_I)$, so we must show that $L(I) = L'$. For the remainder of this proof, we will use $\mathrm{Lf}(u)$ to denote the leaves of u in T and use $\mathrm{Lf}(u) \cap T_I$ to denote the leaves of u in T_I. We use a corresponding notation for vertices $v \in S$.

We claim that $L(I) \subseteq L'$. Let $(u_{red}, v_{red}) \in L(I)$ for some $(u, v) \in L$. By Definition 26 and Lemma 27, $\mathrm{Lf}(u) \cap T_I = \mathrm{Lf}(u_{red}) \cap T_I$ is matched with $\mathrm{Lf}(v) \cap S_{\phi(I)} = \mathrm{Lf}(v_{red}) \cap S_{\phi(I)}$ by $\phi|_I$ and $|\mathrm{Lf}(u_{red}) \cap T_I| = |\mathrm{Lf}(v_{red}) \cap S_{\phi(I)}| > 1$. Also recall from Definition 26 that u_{red} is the minimal $u' \leqslant_T u$ with $\mathrm{Lf}(u) \cap T_I = \mathrm{Lf}(u') \cap T_I$. Note that this u_{red} cannot be a leaf in T since we know $|\mathrm{Lf}(u) \cap T_I)| = |\mathrm{Lf}(u_{red}) \cap T_I| > 1$. Now let T_1 and T_2 be the subtrees of T such that $\mathrm{root}(T_1)$ and $\mathrm{root}(T_2)$ are the children of u_{red}. Then the definition of u_{red} implies that $\mathrm{Lf}(\mathrm{root}(T_1)) \cap T_I$ and $\mathrm{Lf}(\mathrm{root}(T_2)) \cap T_I$ are both proper, nonempty subsets of $\mathrm{Lf}(u_{red}) \cap T_I$, and from this, we conclude that nonempty subtrees $T_1' \subseteq T_1$ and $T_2' \subseteq T_2$ appear in the minimal subtree containing $\{t_i\}_{i \in I}$. This implies that u_{red} must appear in the minimal subtree containing $\{t_i\}_{i \in I}$, as it is the minimal vertex that is an ancestor of the vertices in $T_1' \cup T_2'$. Furthermore, this also implies that u_{red} has a child in T_1' and a child in T_2'. Since u_{red} has two children in the minimal subtree containing $\{t_i\}_{i \in I}$, it will not be suppressed when forming T_I. We conclude that u_{red} is an internal vertex of T_I, and a similar argument shows v_{red} is an internal vertex of $S_{\phi(I)}$. Combined, we see that u_{red} and v_{red} are internal vertices of $(T_I, S_{\phi(I)}, \phi|_I)$, and $\mathrm{Lf}(u_{red}) \cap T_I$ is matched with $\mathrm{Lf}(v_{red}) \cap S_{\phi(I)}$ by $\phi|_I$. Thus, $(u_{red}, v_{red}) \in L'$, and we conclude that $L(I) \subseteq L'$.

To finish proving (b), we must show that $L(I) \supset L'$. Consider any leaf-matched pair $(u', v') \in L'$. Observe that $u' = u'_{red}$ and $v' = v'_{red}$ since u' and v' are vertices in $(T_I, S_{\phi(I)}, \phi|_I)$. During $\texttt{ModifiedUntangle}(T, S, \phi|_I)$, either u' or v' appears first in some partial layout (X_j, Y_j), so we assume without loss of generality that it is $u' \in X_j$. Since v' has not appeared yet in a partial layout, some vertex $y' \geqslant_S v'$ must be in Y_j. Since $\mathrm{Lf}(u') \cap T_I$ is matched with a subset of $\mathrm{Lf}(y') \cap S_{\phi(I)}$ by $\phi|_I$, we know that y' is the unique neighbor of u' in (X_j, E_j, Y_j), and if $\deg(y') \geqslant 1$ in (X_j, E_j, Y_j), then y' will be refined before we refine u'. Eventually, we will obtain some partial layout (X_k, Y_k) where some $w' \geqslant_S v'$ will appear in Y_k with $\deg(w') = 1$ in (X_k, E_k, Y_k), and hence u' and w' are unique neighbors of one another in (X_k, E_k, Y_k). After this occurs, `ModifiedUntangle` will add (u', w') to L before it refines u' or w'. Since $(u', v') \in L'$, we know $\mathrm{Lf}(u') \cap T_I$ is matched with $\mathrm{Lf}(v') \cap S_{\phi(I)}$. Since u' and w' had degree one in (X_k, E_k, Y_k), Lemma 27 implies $\mathrm{Lf}(u') \cap T_I$ is also matched with $\mathrm{Lf}(w') \cap S_{\phi(I)}$ by $\phi|_I$, and therefore $\mathrm{Lf}(v') \cap S_{\phi(I)} = \mathrm{Lf}(w') \cap S_{\phi(I)}$. Since $v'_{red} = v'$, we conclude that $w'_{red} = v'_{red} = v'$, and thus $(u', v') = (u'_{red}, w'_{red}) \in L(I)$. □

4.1 Preserving subtanglegram planarity and reducing crossings

Throughout this subsection, fix (T, S, ϕ) as a tanglegram of size n, and fix $I = [n] \setminus \{i\}$ for some $i \in [n]$. Let $L(I)$ be the leaf-matched pairs of $(T_I, S_{\phi(I)}, \phi|_I)$, and let (X, Y) be a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$. We also use the notation

$$u_0 = \text{parent of } t_i \in T \quad \text{ and } \quad v_0 = \text{parent of } s_{\phi(i)} \in S, \tag{5}$$

which are also the unique internal vertices in (T, S, ϕ) that are not in $(T_I, S_{\phi(I)}, \phi|_I)$. To solve the Tanglegram Single Edge Insertion Problem, we pursue the following questions:

(1) What operations on (X, Y) produce another layout (X', Y') that is also planar when restricted to $(T_I, S_{\phi(I)}, \phi|_I)$?

(2) How can we efficiently find the operation(s) that correspond to a solution to the Tanglegram Single Edge Insertion Problem?

In this section, we answer (1) and establish some results that will lead to (2). A complete answer to (2) requires some cases, which we detail in Section 4.2. Before we answer (1), we start with a definition. While a special case of this definition is sufficient for answering (1), the extra generality will be useful later in Section 5.

Definition 29. Let T be a tree, let $u \in T$ be an internal vertex, and let u_1, u_2 be the children of u. A *subtree switch* (sometimes abbreviated *switch*) at u is the operation on T that interchanges the two subtrees rooted at u_1 and u_2, while maintaining the relative order of all leaves within each subtree. Note that a subtree switch is equivalent to a flip at u and then a flip at each child of u that is not a leaf.

Lemma 30. *Suppose (X, Y) and (X', Y') are both layouts of (T, S, ϕ) that restrict to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$. Then (X', Y') can be obtained from (X, Y) using a sequence of the following operations:*

- *paired flips at $(u, v) \in L(I)$, and*
- *subtree switches at u_0 and v_0.*

Proof. Note that flips at internal vertices $u \in T$ and $v \in S$ generate all trees isomorphic to T and S, and all of these flips commute with one another. Hence, sequences of flips at internal vertices generate all layouts of a tanglegram. From this, we know that some sequence of flips $f_1, \ldots, f_m$ at vertices in T and S maps (X, Y) to (X', Y'), and all of these flips commute with one another. We can assume that we do not flip at any vertex twice, as all flips commute and have order two.

Recall that u_0 is the parent of t_i and v_0 is the parent of $s_{\phi(i)}$. Let u' be the child of u_0 that is not t_i, and let v' be the child of v_0 that is not $s_{\phi(i)}$. Note that these may or may not be internal vertices. If u' is an internal vertex, define g to be a flip at u'. Otherwise, define g as the identity map. Similarly, define h to be a flip at v' or the identity map, respectively corresponding to when v' is an internal vertex or a leaf.

First, suppose that none of the flips $f_1, \ldots, f_m$ are flips at u_0 and v_0. Then we can restrict the layouts (X, Y) and (X', Y') to $(T_I, S_{\phi(I)}, \phi|_I)$ and consider $f_1, \ldots, f_m$ as a sequence of flips in the subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$. Since the restrictions of (X, Y) and

(X', Y') are planar layouts of $(T_I, S_{\phi(I)}, \phi|_I)$, Theorem 1 implies that $f_1, \ldots, f_m$ must be equivalent to a sequence of paired flips at elements in $L(I)$.

Now suppose that some f_j corresponds to a flip at u_0, but no flip at v_0 occurs. Since all flips commute, we can assume without loss of generality that this is f_1. Then another sequence of flips that maps (X, Y) to (X', Y') is

$$f_1, g, g, f_2, \ldots, f_m,$$

as all flips have order 2. The composition $g \circ f_1$ is a subtree switch at u_0 that maps (X, Y) to a layout (X'', Y'') that is also planar when restricted to $(T_I, S_{\phi(I)}, \phi|_I)$. The sequence $g, f_2, \ldots, f_m$ maps (X'', Y'') to (X', Y'), and none of the flips involve the vertices u_0 and v_0. By the preceding paragraph, this sequence must be equivalent to a sequence of paired flips at elements in $L(I)$. We can use a similar argument when some f_j corresponds to a flip at v_0 and no flip at u_0 occurs.

Finally, if flips at both u_0 and v_0 occur, then we assume without loss of generality that these flips are f_1 and f_2, respectively. We then consider the sequence

$$f_1, g, f_2, h, g, h, f_3, \ldots, f_m,$$

which is equivalent to $f_1, \ldots, f_m$ since all flips commute and have order 2. Using similar reasoning, $g \circ f_1$ and $h \circ f_2$ are subtree switches, and $g, h, f_3, \ldots, f_m$ correspond to a sequence of paired flips at elements in $L(I)$. □

Lemma 31. *Let $(u, v) \in L(I)$, and let (X', Y') be the image of (X, Y) after a paired flip at (u, v).*

(a) If $u \not>_T t_i$ and $v \not>_S s_{\phi(i)}$, then (X, Y) and (X', Y') have the exact same crossings.

(b) If $u >_T t_i$ and $v >_S s_{\phi(i)}$, then (X, Y) and (X', Y') have the exact same crossings.

Proof. For (a), we suppose $u \not>_T t_i$ and $v \not>_S s_{\phi(i)}$. Suppose $(t_i, s_{\phi(i)})$ crosses some but not all of the edges between $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$. Then t_i appears in the middle of the leaves in $\mathrm{Lf}(u)$, or $s_{\phi(i)}$ appears in the middle of the leaves in $\mathrm{Lf}(v)$. These respectively contradict the assumptions $u \not>_T t_i$ and $v \not>_S s_{\phi(i)}$, as the leaves in $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$ must appear consecutively in any layout. From this, we conclude that $(t_i, s_{\phi(i)})$ either crosses all or none of the edges between $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$. Then a paired flip at (u, v) does not affect any crossings.

For (b), suppose that for the pair (u, v), both $u >_T t_i$ and $v >_S s_{\phi(i)}$. Then all crossings of (T, S, ϕ) are contained in the subtanglegram with trees rooted at u and v. A paired flip at (u, v) reflects this subtanglegram, preserving all of the crossings. Visualizations of these arguments are shown in Figure 10. □

The preceding lemmas suggest that we should focus on subtree switches at u_0 and v_0, as well as the leaf-matched pairs $(u, v) \in L(I)$ where exactly one of $u >_T t_i$ or $v >_S s_{\phi(i)}$ is true. Motivated by these observations, we define the following sets:

$$\begin{aligned} L(I)_T &= \{(u, v) \in L(I) : u >_T t_i, v \not>_S s_{\phi(i)}\}, \\ L(I)_S &= \{(u, v) \in L(I) : u \not>_T t_i, v >_S s_{\phi(i)}\}. \end{aligned} \tag{6}$$

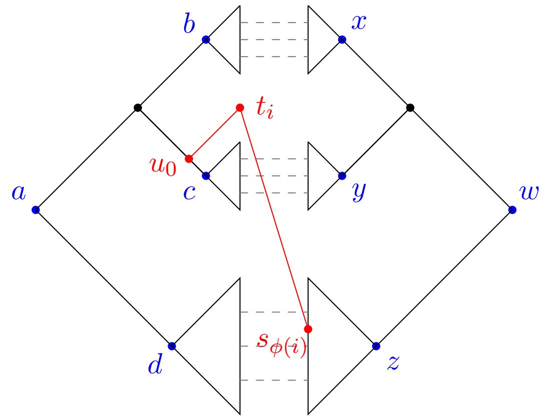

Figure 10: Lemma 31 shows that paired flips at $(a, w), (v, x), (c, y)$ do not change any crossings. Notice that a paired flip at (d, z) affects crossings involving the edges between $\mathrm{Lf}(d)$ *and* $\mathrm{Lf}(z)$*.*

The two preceding lemmas then imply the following result concerning solutions to the Single Edge Insertion Problem.

Corollary 32. *A solution to the Tanglegram Single Edge Insertion Problem can be obtained by starting at (X, Y) and performing a sequence of the following operations:*

- *paired flips at $(u, v) \in L(I)_T \cup L(I)_S$, and*
- *subtree switches at u_0 and v_0.*

Proof. Let (X_{min}, Y_{min}) correspond to a solution of the Tanglegram Single Edge Insertion Problem. By Lemma 30, we can obtain (X_{min}, Y_{min}) from (X, Y) using a sequence $f_1, f_2, \ldots, f_m$ of paired flips at $(u, v) \in L(I)$ and subtree switches at u_0 and v_0. Some of these f_j may correspond to paired flips at $(u, v) \in L(I)$ where both or neither of $u >_T t_i$, $v >_S s_{\phi(i)}$ are true. We can assume these paired flips correspond to $f_{k+1}, \ldots, f_m$ for some $0 \leqslant k \leqslant m$, as all flips commuting implies that paired flips and subtree switches also commute. By Lemma 31 with induction, if we only perform $f_1, f_2, \ldots, f_k$, then we obtain another layout (X'_{min}, Y'_{min}) that has the exact same crossings as (X_{min}, Y_{min}). Thus, (X'_{min}, Y'_{min}) also solves the Tanglegram Single Edge Insertion Problem, and the result follows. □

From Corollary 32, we see that solving the Tanglegram Single Edge Insertion Problem reduces to finding specific sequences of paired flips at $(u, v) \in L(I)_T \cup L(I)_S$ and subtree switches at u_0, v_0. Identifying the correct sequences will require some cases. Before we consider these cases, note that the sets $\{u \in T : (u, v) \in L(I)_T \text{ for some } v \in S\}$ and $\{v \in S : (u, v) \in L(I)_S \text{ for some } u \in T\}$ are linearly ordered since they form subsets of the ancestors of t_i and $s_{\phi(i)}$, respectively. In particular, this implies that $u >_T u_0$ for all $(u, v) \in L(I)_T$, and $v >_S v_0$ for all $(u, v) \in L(I)_S$ since u_0 and v_0 are respectively the parents of t_i and $s_{\phi(i)}$. Additionally, whenever $L(I)_T$ and $L(I)_S$ are nonempty, each set has a unique "maximal" element, which we denote

$$\begin{aligned} (u_{Tmax}, v_{Tmax}) &= \text{unique } (u,v) \in L(I)_T \text{ such that } u \geqslant_T u' \text{ for all } (u',v') \in L(I)_T, \\ (u_{Smax}, v_{Smax}) &= \text{unique } (u,v) \in L(I)_S \text{ such that } v \geqslant_S v' \text{ for all } (u',v') \in L(I)_S. \end{aligned} \quad (7)$$

Some examples are shown in Figure 11. We conclude this section with one final lemma describing the implications of ancestry relations between u_0 and u_{Smax}, or v_0 and v_{Tmax}.

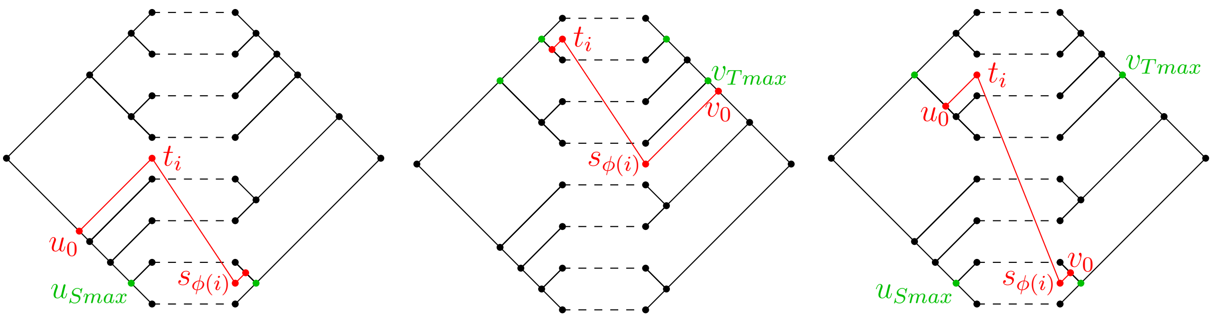

Figure 11: Examples of u_{Smax} and v_{Tmax}.

Lemma 33. *The following properties hold when the appropriate vertex u_{Smax} or v_{Tmax} exists.*

(i) Suppose $u_0 >_T u_{Smax}$, and let $(u,v) \in L(I)$. If $u >_T t_i$, then $v >_S s_{\phi(i)}$. Hence, $L(I)_T = \emptyset$.

(ii) Suppose $u_0 \not>_T u_{Smax}$. Then $u_0 \not>_T u$ for any $(u,v) \in L(I)_S$.

(iii) Suppose $v_0 >_S v_{Tmax}$, and let $(u,v) \in L(I)$. If $v >_S s_{\phi(i)}$, then $u >_T t_i$. Hence, $L(I)_S = \emptyset$.

(iv) Suppose $v_0 \not>_S v_{Tmax}$. Then $v_0 \not>_S v$ for any $(u,v) \in L(I)_T$.

Proof. For (i), suppose $u_0 >_T u_{Smax}$, and consider $(u,v) \in L(I)$ with $u >_T t_i$. Since u_0 is the parent of t_i, this implies that $u >_T u_0$. Combined with the assumption, we conclude

$$u >_T u_0 >_T u_{Smax}.$$

Then in the induced subtree T_I, we know $u >_{T_I} u_{Smax}$. This implies a corresponding relation for their leaf-matched vertices v and v_{Smax} in the induced subtree $S_{\phi(I)}$, so $v >_{S_{\phi(I)}} v_{Smax}$. Since the ancestry relations on $S_{\phi(I)}$ are restrictions of the relations on S, this implies $v >_S v_{Smax}$. We know $v_{Smax} >_S s_{\phi(i)}$ by the definition of $L(I)_S$, so we conclude that

$$v >_S v_{Smax} >_S s_{\phi(i)}.$$

For (ii), suppose $u_0 \not>_T u_{Smax}$. Assume by contradiction that $u_0 >_T u$ for some $(u,v) \in L(I)_S$. Then by the definition of (u_{Smax}, v_{Smax}), we know that $v_{Smax} >_S v$, and therefore $u_{Smax} >_T u$ by the same argument involving T_I and $S_{\phi(I)}$ in the proof of (i) above. We see that u_{Smax} and u_0 are both ancestors of u, and the assumption $u_0 \not>_T u_{Smax}$ then implies

$$u_{Smax} >_T u_0 >_T u$$

since the ancestors of u are linearly ordered. However, u_0 is the parent of t_i, so this implies $u_{Smax} >_T t_i$, contradicting $(u_{Smax}, v_{Smax}) \in L(I)_S$. Results (iii) and (iv) follow by similar arguments. □

4.2 Insertion Algorithm

In the previous section, we established that a solution to the Single Edge Insertion Problem only requires us to consider $L(I)_T, L(I)_S, u_0$, and v_0. Now we wish to efficiently find a sequence of paired flips and subtree switches that solves the Tanglegram Single Edge Insertion Problem. Our approach for finding such a sequence will be different depending on where u_0 and v_0 are inserted. The three cases will be $u_0 >_T u_{Smax}$, $v_0 >_S v_{Tmax}$, or when neither of these is true, and an example of each of case was previously shown in Figure 11. In this subsection, we solve the three different cases, and then we will combine our results to form the `Insertion` algorithm that solves the Tanglegram Single Edge Insertion Problem.

We start with the case $u_0 >_T u_{Smax}$. Lemma 33 implies $L(I)_T$ is empty, so we can focus our attention on v_0, u_0, and $L(I)_S = \{(u_j, v_j)\}_{j=1}^m$, where elements have been indexed so that $v_1 <_S v_2 <_S \ldots <_S v_m$. Below we give an algorithm for this first case of single edge insertion. Notice that we do not consider a flip or switch at any vertex until we have considered all ancestors of that vertex, and we make these flip and switch choices based on edges in a set we call $E(u_0), E(v_0)$, and $E(v_j)$ for $j = 1, 2, \ldots, m$. These sets track the crossings that can be affected by an operation at that vertex and cannot be affected by operations at descendants of that vertex. An example of the algorithm is shown in Figure 12.

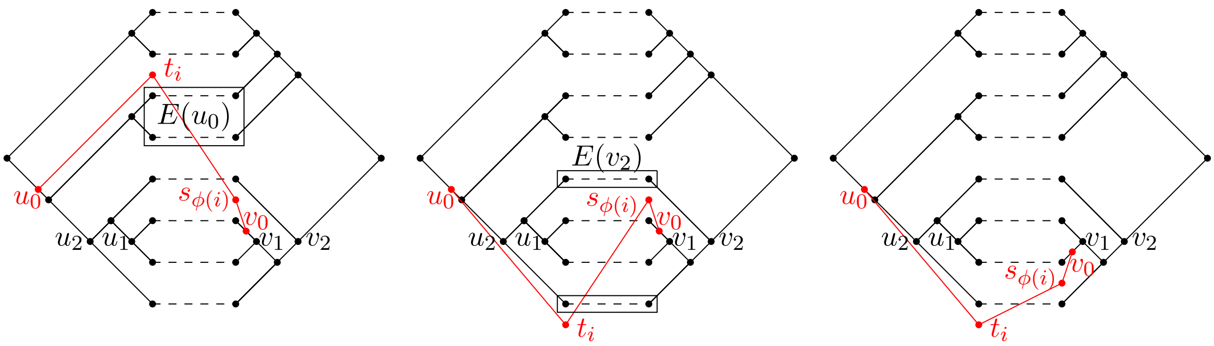

Figure 12: If `ModifiedUntangle`$(T, S, \phi|_I)$ *returns the layout on the left, then Algorithm 3 would first perform a subtree switch at* u_0 *to obtain the layout in the middle. Then it would not perform a paired flip at* (u_2, v_2)*, would perform a paired flip at* (u_1, v_1)*, and would not perform a subtree switch at* v_0*, returning the layout on the right with one crossing.*

Algorithm 3: `Insertion Case` $u_0 >_T u_{Smax}$

Input: tanglegram (T, S, ϕ), index i s.t. $(T_I, S_{\phi(I)}, \phi|_I)$ is planar for $I = [n] \setminus \{i\}$
Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

`// Step 1: initialize the algorithm.`

$(X, Y), L \coloneqq$ `ModifiedUntangle`$(T, S, \phi|_I)$
construct $L(I)$ from L using Definition 26
$u_0 \coloneqq$ parent of t_i, $v_0 \coloneqq$ parent of $s_{\phi(i)}$
$L(I)_S \coloneqq \{(u, v) \in L(I) : u \not>_T t_i, v >_S s_{\phi(i)}\}$

`// Step 2: construct edge sets.`

linearly order $L(I)_S = \{(u_j, v_j)\}_{j=1}^m$ so that $v_1 <_S v_2 <_S \ldots <_S v_m$
$E(u_0) \coloneqq$ between-tree edges with an endpoint in $\mathrm{Lf}(u_0) \setminus \mathrm{Lf}(u_m)$
$E(v_0) \coloneqq$ between-tree edges with an endpoint in $\mathrm{Lf}(v_0) \setminus \{s_{\phi(i)}\}$
for $j = 1, 2, \ldots, m$, **do**
  $E(v_j) \coloneqq$ between-tree edges with an endpoint in $\mathrm{Lf}(v_j) \setminus \mathrm{Lf}(v_{j-1})$

`// Step 3: use paired flips and subtree switches.`

if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(u_0)$ *in the layout* (X, Y), **then**
  update $X \coloneqq$ `SubtreeSwitch`(X, u_0)
for $j = m, \ldots, 2, 1$, **do**
  if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(v_j)$ *in the layout* (X, Y), **then**
    update $(X, Y) \coloneqq$ `PairedFlip`$((X, Y), (u_j, v_j))$
if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(v_0)$ *in the layout* (X, Y), **then**
  update $Y \coloneqq$ `SubtreeSwitch`(Y, v_0)
return (X, Y)

Lemma 34. *Algorithm 3 solves the Tanglegram Single Edge Insertion Problem when* $L(I)_S \neq \emptyset$ *and* $u_0 >_T u_{Smax}$.

Proof. By Lemma 28, the initial layout (X, Y) of (T, S, ϕ) in Step 1 of Algorithm 3 restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$, and $L(I)$ is the set of leaf-matched pairs of $(T_I, S_{\phi(I)}, \phi|_I)$. The assumption $u_0 >_T u_{Smax}$ with Lemma 33 implies $L(I)_T = \emptyset$. Corollary 32 with $L(I)_T = \emptyset$ implies that starting with (X, Y), some sequence of paired flips at $(u_j, v_j) \in L(I)_S$ and subtree switches at u_0 and v_0 solves the Tanglegram Single Edge Insertion Problem. The algorithm makes choices in the order $u_0, (u_m, v_m), \ldots, (u_1, v_1), v_0$, so we will show that our choice at each step can extend to a solution to the Single Edge Insertion Problem, and thus the algorithm terminates at a solution.

The algorithm starts with u_0. Some operations at $u_0, (u_j, v_j) \in L(I)_S$, and v_0 produce a solution (X_{min}, Y_{min}) to the Single Edge Insertion Problem. This layout contains sublists for the elements in $\mathrm{Lf}(u_0)$ and the leaves paired with them, which have the form either $(t_i X_1 U X_2, Y_1 V Y_2)$ or $(X_1 U X_2 t_i, Y_1 V Y_2)$, where U, V is an ordering of $\mathrm{Lf}(u_m), \mathrm{Lf}(v_m)$ and X_1, X_2, Y_1, Y_2 order the remaining leaves. We focus on the case $(t_i X_1 U X_2, Y_1 V Y_2)$

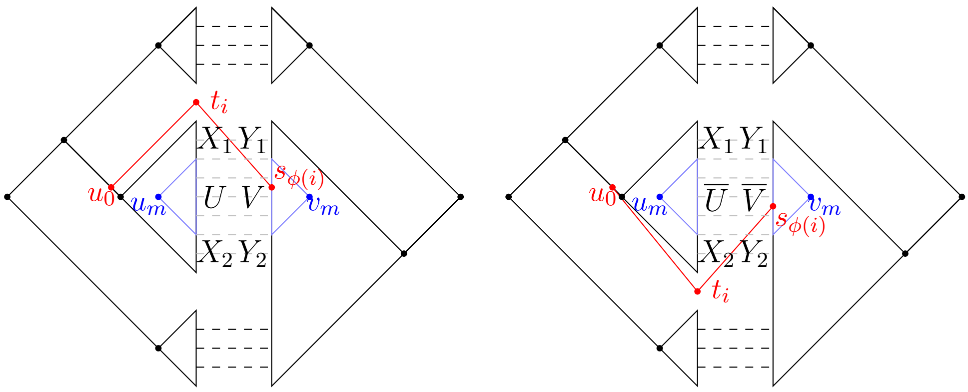

Figure 13: The effect of a subtree switch at u_0 and a paired flip at (u_m, v_m).

illustrated in Figure 13, as the other case follows by similar reasoning. Notice that $E(u_0) = \text{Edges}(X_1) \cup \text{Edges}(X_2)$, where $\text{Edges}(X_i)$ is the set of between-tree edges with a leaf in X_i.

Beginning with $(t_i X_1 U X_2, Y_1 V Y_2)$, if we perform a subtree switch at u_0, notice that we can also perform a paired flip at (u_m, v_m) to obtain

$$(t_i X_1 U X_2, Y_1 V Y_2) \xrightarrow{u_0 \text{ subtree switch}} (X_1 U X_2 t_i, Y_1 V Y_2) \xrightarrow{(u_m, v_m) \text{ paired flip}} (X_1 \overline{U} X_2 t_i, Y_1 \overline{V} Y_2),$$

which is also illustrated in Figure 13. We call this new layout (X', Y'). Notice that crossings between $(t_i, s_{\phi(i)})$ and $\text{Edges}(U)$ are the same in (X_{min}, Y_{min}) and (X', Y'), so the choice of subtree switch at u_0 does not prevent minimization of crossings in $\text{Edges}(U)$. However, while $(t_i, s_{\phi(i)})$ crosses $\text{Edges}(X_1)$ in (X_{min}, Y_{min}), it crosses $\text{Edges}(X_2)$ in the layout (X', Y'), and these crossings cannot be affected by choices in $L(I)_S \cup \{v_0\}$. Thus, the choice at u_0 that extends to a solution to the Tanglegram Single Edge Insertion Problem must be one that minimizes crossings in $E(u_0) = \text{Edges}(X_1) \cup \text{Edges}(X_2)$, so the algorithm's choice at u_0 in lines 10-11 extends to a solution.

Now consider paired flips $(u_j, v_j) \in L(I)_S$. Suppose that we made choices that extend to a solution to the Single Edge Insertion Problem at u_0 and any $(u_{j'}, v_{j'}) \in L(I)_S$ with $v_{j'} >_S v_j$. Then we know that some choices at $(u_j, v_j), \ldots, (u_1, v_1)$, and v_0 lead to a solution to the Single Edge Insertion Problem. This solution contains an ordering of $\text{Lf}(u_j), \text{Lf}(v_j)$. If $j \geqslant 2$, denote this ordering as $(X_1 U X_2, Y_1 V Y_2)$ with U, V an ordering of $\text{Lf}(u_{j-1}), \text{Lf}(v_{j-1})$ as shown on the left in Figure 14. If we perform a paired flip at (u_j, v_j), then we can also perform a paired flip at (u_{j-1}, v_{j-1}) to obtain

$$(X_1 U X_2, Y_1 V Y_2) \xrightarrow{(u_j, v_j) \text{ paired flip}} (\overline{X_2}\,\overline{U}\,\overline{X_1}, \overline{Y_2}\,\overline{V}\,\overline{Y_1}) \xrightarrow{(u_{j-1}, v_{j-1}) \text{ paired flip}} (\overline{X_2} U \overline{X_1}, \overline{Y_2} V \overline{Y_1}).$$

If $j = 1$, the ordering of $\text{Lf}(u_1), \text{Lf}(v_1)$ is either $(X_0, Y_1 s_{\phi(i)} V Y_2)$ or $(X_0, Y_1 V s_{\phi(i)} Y_2)$, where V is an ordering of $\text{Lf}(v_0) \setminus \{s_{\phi(i)}\}$. Performing a paired flip at (u_1, v_1) and a subtree switch at v_0 respectively results in $(\overline{X}_0, \overline{Y_2} s_{\phi(i)} \overline{V}\,\overline{Y_1})$ or $(\overline{X}_0, \overline{Y_2}\,\overline{V} s_{\phi(i)} \overline{Y_1})$.

In all cases, choosing to perform a paired flip at (u_j, v_j) does not prevent minimization of crossings between $(t_i, s_{\phi(i)})$ and $\text{Edges}(V)$. Thus, the choice at (u_j, v_j) that extends to

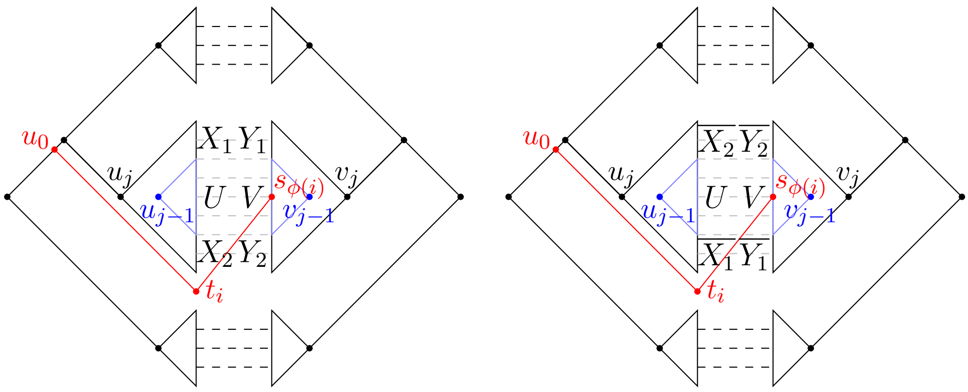

Figure 14: The effect of paired flips at (u_j, v_j) and (u_{j-1}, v_{j-1}) when $2 \leqslant j \leqslant m$.

a solution must be the one that minimizes crossings in $E(v_j) = \text{Edges}(Y_1) \cup \text{Edges}(Y_2)$, as these crossings cannot be affected by operations at $(u_{j-1}, v_{j-1}), \ldots, (u_1, v_1)$, and v_0. Then the algorithm's choice at (u_j, v_j) in lines 13-14 extends to a solution. Using induction, we conclude that the sequence of paired flips at elements in $L(I)_S$ extends to a solution.

Finally, for v_0, we must minimize crossings involving $(t_i, s_{\phi(i)})$ and between-tree edges with an endpoint in $\text{Lf}(v_0)$. This is precisely what the algorithm does with the set $E(v_0)$. Combined, we conclude that Algorithm 3 solves the Tanglegram Single Edge Insertion Problem when $u_0 >_T u_{Smax}$. □

Remark 35. When we consider whether or not to perform a subtree switch at u_0, it is possible that $|X_1| = |X_2|$, and in this case, both choices at u_0 extend to a solution to the Tanglegram Single Edge Insertion Problem. Similarly, when $|Y_1| = |Y_2|$, both choices of whether or not to perform a paired flip at $(u_j, v_j) \in L(I)_S$ extend to a solution. In these situations, we choose not to perform the subtree switch or paired flip for efficiency reasons.

Remark 36. One can analogously construct an `Insertion Case` $v_0 >_S v_{Tmax}$ algorithm for the situation $L(I)_T \neq \emptyset$ and $v_0 >_S v_{Tmax}$. We include the algorithm in the Appendix as Algorithm 8. Proof of its effectiveness follows from a similar argument to the one for Lemma 34.

We now consider the remaining case, where $u_0 \not>_T u_{Smax}$ and $v_0 \not>_S v_{Tmax}$. In this case, Lemma 33 implies $u_0 \not>_T u$ for any $(u, v) \in L(I)_S$ and $v_0 \not>_S v$ for any $(u, v) \in L(I)_T$. We linearly order $L(I)_S = \{(u_j, v_j)\}_{j=1}^{k}$ and $L(I)_T = \{(u_j, v_j)\}_{j=k+1}^{k+m}$ so that $v_0 <_S v_1 <_S v_2 <_S \ldots <_S v_k$ and $u_0 <_T u_{k+1} <_T u_{k+2} <_T \ldots <_T u_{k+m}$. Then we define the E sets in a similar way as before. For $j \in \{1, 2, \ldots, k\}$, define

$$E(v_j) \coloneqq \text{ between-tree edges with an endpoint in } \text{Lf}(v_j) \setminus \text{Lf}(v_{j-1}),$$

and for $j \in \{k+2, \ldots, k+m\}$, define

$$E(u_j) \coloneqq \text{ between-tree edges with an endpoint in } \text{Lf}(u_j) \setminus \text{Lf}(u_{j-1}).$$

Finally, define

$$E(u_{k+1}) \coloneqq \text{ between-tree edges with an endpoint in } \text{Lf}(u_{k+1}) \setminus \text{Lf}(u_0),$$

$$E(v_0) := \text{ between-tree edges with an endpoint in } \mathrm{Lf}(v_0) \setminus \{s_{\phi(i)}\},$$

$$E(u_0) := \text{ between-tree edges with an endpoint in } \mathrm{Lf}(u_0) \setminus \{t_i\}.$$

The next lemma gives us a key property concerning these sets. We wish to use the E sets to minimize crossings, and this lemma will show that we do not need to worry about these sets intersecting in most cases.

Lemma 37. *Suppose $u_0 \not>_T u_{Smax}$ and $v_0 \not>_S v_{Tmax}$. Then $E(u_j) \cap E(v_\ell) \neq \emptyset$ can only occur when $j = \ell = 0$.*

Proof. Suppose $E(u_j) \cap E(v_\ell) \neq \emptyset$ for some $j, \ell \neq 0$, that is, there exists $(u_j, v_j) \in L(I)_T$, $(u_\ell, v_\ell) \in L(I)_S$, and a between-tree edge (t, s) such that $t \in \mathrm{Lf}(u_j) \setminus \mathrm{Lf}(u_{j-1})$ and $s \in \mathrm{Lf}(v_\ell) \setminus \mathrm{Lf}(v_{\ell-1})$. The assumption $t \in \mathrm{Lf}(u_j) \setminus \mathrm{Lf}(u_{j-1})$ implies $u_j >_T t$. Since $(u_j, v_j) \in L(I)_T$, the definition of $L(I)_T$ implies that $u_j >_T t_i$ and $v_j \not>_S s_{\phi(i)}$. Furthermore, (u_j, v_j) is a leaf-matched pair of $(T_I, S_{\phi(I)}, \phi|_I)$, so $u_j >_T t$ implies $v_j >_S s$. Combined, we see that $u_j >_T t$, $u_j >_T t_i$, $v_j >_S s$, and $v_j \not>_S s_{\phi(i)}$.

Using similar reasoning, the assumption $s \in \mathrm{Lf}(v_\ell) \setminus \mathrm{Lf}(v_{\ell-1})$ implies $v_\ell >_S s$, and $(u_\ell, v_\ell) \in L(I)_S$ implies $v_\ell >_S s_{\phi(i)}$. Since the ancestors of s are linearly ordered, the fact that v_ℓ is an ancestor of both s and $s_{\phi(i)}$ while v_j is only an ancestor of s implies $v_\ell >_S v_j$, giving us the situation illustrated in Figure 15. This implies $u_\ell >_T u_j >_T t_i$, which contradicts $(u_\ell, v_\ell) \in L(I)_S$.

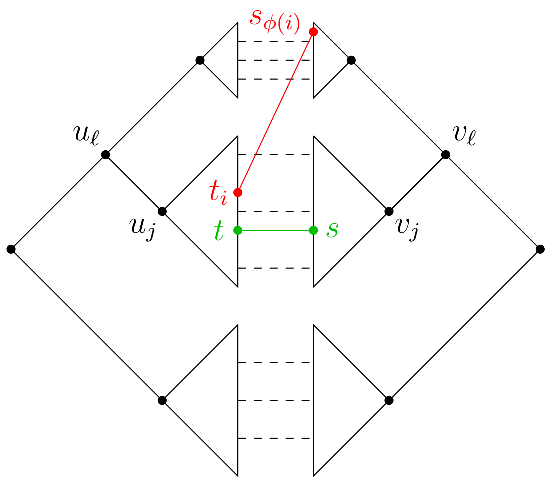

Figure 15: The proof of Lemma 37 (i).

Now suppose that $E(u_0) \cap E(v_j) \neq \emptyset$ for some $j \neq 0$, that is, there exists some $(u_j, v_j) \in L(I)_S$ and some between-tree edge (t, s) with $t \in \mathrm{Lf}(u_0) \setminus \{t_i\}$ and $s \in \mathrm{Lf}(v_j) \setminus \mathrm{Lf}(v_{j-1})$. Since $t \in \mathrm{Lf}(u_0) \setminus \{t_i\}$, we see that $u_0 >_T t$. Using the fact that (u_j, v_j) is a leaf-matched pair of $(T_I, S_{\phi(I)}, \phi|_I)$, $v_j >_S s$ implies $u_j >_T t$. Since ancestors of t are linearly ordered, either $u_j >_T u_0$ or $u_j <_T u_0$. Using our assumption $u_0 \not>_T u_{Smax}$ with Lemma 33, we conclude that $u_j >_T u_0 >_T t_i$, which contradicts $(u_j, v_j) \in L(I)_S$. The case $E(u_j) \cap E(v_0) \neq \emptyset$ for $j \neq 0$ is ruled out by similar reasoning. $\square$

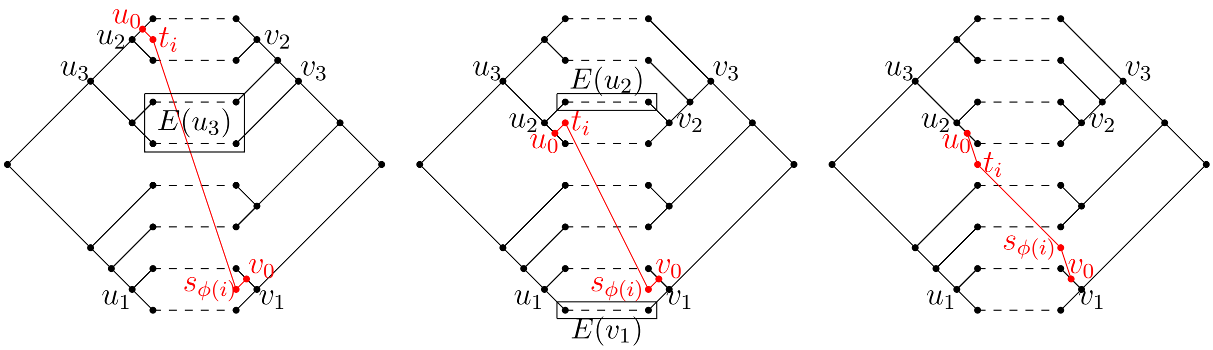

Figure 16: If the layout from the left is the output of `ModifiedUntangle`$(T, S, \phi|_I)$*, then Algorithm 4 will first perform a paired flip at* (u_3, v_3)*. Afterwards, it will not perform a paired flip at* (u_2, v_2) *or* (u_1, v_1)*. Since* $E(u_0) \cap E(v_0) = \emptyset$*, the algorithm will perform subtree switches at* u_0 *and* v_0*, returning the layout shown on the right.*

We now define an algorithm to solve the remaining cases of the Tanglegram Single Edge Insertion Problem. As before, we consider flips at all ancestors of a vertex before considering the vertex itself, and make flip and switch choices according to crossings in the $E(u_j)$ and $E(v_j)$ sets. An example of the algorithm is shown in Figure 16.

Lemma 38. *Algorithm 4 solves the Tanglegram Single Edge Insertion Problem when* $u_0 \not>_T u_{Smax}$ *and* $v_0 \not>_S v_{Tmax}$*, where we also consider* $L(I)_S = \emptyset$ *and* $L(I)_T = \emptyset$*, respectively, as cases of* $u_0 \not>_T u_{Smax}$ *and* $v_0 \not>_S v_{Tmax}$*.*

Proof. Using the same argument as in Lemma 34, starting at (X, Y) in Step 1 and performing some sequence of paired flips at elements in $L(I)_T \cup L(I)_S$ and subtree switches at $\{u_0, v_0\}$ solves the Tanglegram Single Edge Insertion Problem. Algorithm 4 first considers the elements of $L(I)_T$ in the order $(u_{k+m}, v_{k+m}), \ldots, (u_{k+1}, v_{k+1})$ and then elements of $L(I)_S$ in the order $(u_k, v_k), \ldots, (u_1, v_1)$. Similar to Lemma 34, we show that our choice at each element of $L(I)_T$ and $L(I)_S$ in this order extends to a solution to the Tanglegram Single Edge Insertion Problem.

Consider $(u_j, v_j) \in L(I)_T$, and assume that our choices at all previously considered vertices extend to a solution to the Single Edge Insertion Problem. Some choice of operations at $(u_{j'}, v_{j'}) \in L(I)_T$ with $u_{j'} \leqslant_T u_j$, $(u_\ell, v_\ell) \in L(I)_S$, u_0, and v_0 solves the Single Edge Insertion Problem. This gives us a layout (X_{min}, Y_{min}) that includes an ordering of $\mathrm{Lf}(u_j), \mathrm{Lf}(v_j)$. When $j \neq k+1$, this ordering has the form $(X_1 U X_2, Y_1 V Y_2)$ with U, V an ordering of $\mathrm{Lf}(u_{j-1}), \mathrm{Lf}(v_{j-1})$ as shown in Figure 17. If we perform a paired flip at (u_j, v_j), then we can perform a paired flip at (u_{j-1}, v_{j-1}) to obtain the ordering $(\overline{X_2} U \overline{X_1}, \overline{Y_2} V \overline{Y_1})$ on $\mathrm{Lf}(u_j), \mathrm{Lf}(v_j)$. Letting $\mathrm{Edges}(U)$ be the between-tree edges with an endpoint in U, we see that the original layout (X_{min}, Y_{min}) and the new layout have the same crossings between $(t_i, s_{\phi(i)})$ and $\mathrm{Edges}(U)$. Thus, regardless of our choice at (u_j, v_j), we can minimize crossings in $\mathrm{Edges}(U)$.

When $j = k+1$, the original ordering has the form $(X_1 t_i U X_2, Y_0)$ or $(X_1 U t_i X_2, Y_0)$

Algorithm 4: `Insertion Case` $u_0 \not>_T u_{Smax}$ `and` $v_0 \not>_S v_{Tmax}$

Input: tanglegram (T, S, ϕ), index i s.t. $(T_I, S_{\phi(I)}, \phi|_I)$ is planar for $I = [n] \setminus \{i\}$
Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

`// Step 1: initialize the algorithm.`

$(X, Y), L := \texttt{ModifiedUntangle}(T, S, \phi|_I)$
construct $L(I)$ from L using Definition 26
$u_0 :=$ parent of t_i, $v_0 :=$ parent of $s_{\phi(i)}$
$L(I)_S := \{(u, v) \in L(I) : u \not>_T t_i, v >_S s_{\phi(i)}\}$
$L(I)_T := \{(u, v) \in L(I) : u >_T t_i, v \not>_S s_{\phi(i)}\}$

`// Step 2: construct edge sets.`

linearly order $L(I)_S = \{(u_j, v_j)\}_{j=1}^{k}$ so that $v_1 <_S v_2 <_S \ldots <_S v_k$
linearly order $L(I)_T = \{(u_j, v_j)\}_{j=k+1}^{k+m}$ so that $u_{k+1} <_T u_{k+2} <_T \ldots <_T u_{k+m}$
for $j = 1, 2, \ldots, k,$ **do**
  $E(v_j) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(v_j) \setminus \mathrm{Lf}(v_{j-1})$
for $j = k+2, k+3, \ldots, m,$ **do**
  $E(u_j) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(u_j) \setminus \mathrm{Lf}(u_{j-1})$
$E(u_{k+1}) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(u_{k+1}) \setminus \mathrm{Lf}(u_0)$
$E(v_0) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(v_0) \setminus \{s_{\phi(i)}\}$
$E(u_0) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(u_0) \setminus \{t_i\}$

`// Step 3: use paired flips in` $L(I)_T$ `and` $L(I)_S$`.`

for $j = k+m, \ldots, k+2, k+1,$ **do**
  if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(u_j)$ *in the layout* (X, Y), **then**
    update $(X, Y) := \texttt{PairedFlip}((X, Y), (u_j, v_j))$
for $j = k, \ldots, 2, 1,$ **do**
  if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(v_j)$ *in the layout* (X, Y), **then**
    update $(X, Y) := \texttt{PairedFlip}((X, Y), (u_j, v_j))$

`// Step 4: use subtree switches at` u_0 `and` v_0`.`

if $E(u_0) \cap E(v_0) = \emptyset$, **then**
  if $(t_i, s_{\phi(i)})$ *crosses the edges in* $E(u_0)$, **then**
    update $X := \texttt{SubtreeSwitch}(X, u_0)$
  if $(t_i, s_{\phi(i)})$ *crosses the edges in* $E(v_0)$, **then**
    update $Y := \texttt{SubtreeSwitch}(Y, v_0)$
  return X, Y
else
  $X', Y' := \texttt{SubtreeSwitch}(X, u_0), \texttt{SubtreeSwitch}(Y, v_0)$
  return a layout in $\{(X, Y), (X', Y), (X, Y'), (X', Y')\}$ with fewest crossings

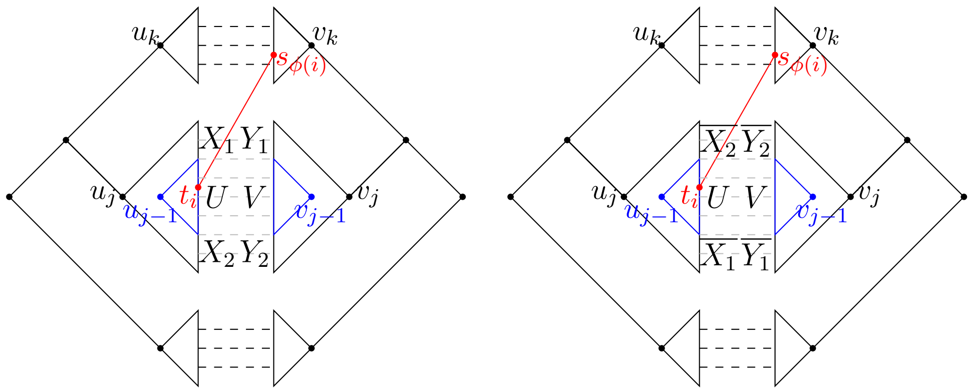

Figure 17: The effect of a paired flip at (u_j, v_j) and (u_{j-1}, v_{j-1}) when $j \geqslant 2$.

with U an ordering of $\mathrm{Lf}(u_0) \setminus \{t_i\}$. If we make a paired flip at (u_{k+1}, v_{k+1}), we can perform a subtree switch at u_0 to respectively obtain $(\overline{X_2} t_i \overline{U}\, \overline{X_1}, \overline{Y_0})$ or $(\overline{X_2}\, \overline{U} t_i \overline{X_1}, \overline{Y_0})$. In either case, the resulting layout will again have the same crossings between $(t_i, s_{\phi(i)})$ and $\mathrm{Edges}(U)$, so regardless of our choice at (u_{k+1}, v_{k+1}), we can still minimize crossings in $\mathrm{Edges}(U)$.

In both of the above cases, notice that the crossings involving $E(u_j) = \mathrm{Edges}(X_1) \cup \mathrm{Edges}(X_2)$ are affected, as $(t_i, s_{\phi(i)})$ will either cross $\mathrm{Edges}(X_1)$ or $\mathrm{Edges}(X_2)$. These crossings cannot be affected by paired flips at any $(u_{j'}, v_{j'}) \in L(I)_T$ with $j' < j$, nor by a subtree switch at u_0. Additionally, by Lemma 37, $E(u_j)$ does not intersect $E(v_\ell)$ for $(u_\ell, v_\ell) \in L(I)_S$, nor does it intersect $E(v_0)$. Then $E(u_j)$ does not contain any edges with a leaf in $\mathrm{Lf}(v_\ell)$ for $\ell = 0, 1, \ldots, k$, and crossings in $E(u_j)$ cannot be affected by any of the choices at $(u_\ell, v_\ell) \in L(I)_S$ and v_0. Combined, we conclude that the choice at (u_j, v_j) that extends to a solution to the Single Edge Insertion Problem must be the one that minimizes crossings in $E(u_j)$, which is precisely what the algorithm does in lines 16-17. Using induction, the sequence of choices in $L(I)_T$ extends to a solution.

Using a similar argument, we obtain the same conclusion for the choices of paired flips in $L(I)_S$. After making appropriate choices of paired flips for elements in $L(I)_T$ and $L(I)_S$, we know some combination of subtree switches at $\{u_0, v_0\}$ solves the Tanglegram Single Edge Insertion Problem. When $E(u_0) \cap E(v_0) = \emptyset$, the choices at u_0 and v_0 affect different crossings, and the algorithm chooses to minimize crossings in each set. When $E(u_0) \cap E(v_0) \neq \emptyset$, the algorithm checks all four possible layouts in lines 28-29 and returns a layout with minimal crossings. In all cases, the output must be a solution to the Tanglegram Single Edge Insertion Problem. □

Using these algorithms and results, we now define a combined algorithm that solves the Tanglegram Single Edge Insertion Problem below. The initial steps are similar to the previous algorithms. We then check for the conditions corresponding to each case, and proceed appropriately.

Proof of Theorem 3. Effectiveness of the `Insertion` Algorithm follows from the cases considered in Step 2 combined with Lemma 34, Remark 36, and Lemma 38, so it remains

Algorithm 5: `Insertion`

Input: tanglegram (T, S, ϕ), index i s.t. $(T_I, S_{\phi(I)}, \phi|_I)$ is planar for $I = [n] \setminus \{i\}$
Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

`// Step 1: initialize the algorithm.`

1 $(X, Y), L \coloneqq$ `ModifiedUntangle`$(T, S, \phi|_I)$
2 construct $L(I)$ from L using Definition 26
3 $u_0 \coloneqq$ parent of t_i, $v_0 \coloneqq$ parent of $s_{\phi(i)}$
4 $L(I)_S \coloneqq \{(u, v) \in L(I) : u \not>_T t_i, v >_S s_{\phi(i)}\}$
5 $L(I)_T \coloneqq \{(u, v) \in L(I) : u >_T t_i, v \not>_S s_{\phi(i)}\}$

`// Step 2: consider cases.`

6 use (6) to define u_{Smax} when $L(I)_S \neq \emptyset$ and v_{Tmax} when $L(I)_T \neq \emptyset$
7 **if** $L(I)_S \neq \emptyset$ *and* $u_0 >_T u_{Smax}$, **then**
8   proceed from Step 2 of Algorithm 3
9 **else if** $L(I)_T \neq \emptyset$ *and* $v_0 >_S v_{Tmax}$, **then**
10   proceed from Step 2 Algorithm 8
11 **else**
12   proceed from Step 2 of Algorithm 4

to show that the algorithm runs in $O(n^2)$ time and space. In Step 1, `ModifiedUntangle` runs in $O(n^2)$ time and space by Remark 17. Additionally, constructing $L(I)$ takes $O(n^2)$ time and space, as L has size at most n, and for every $(u, v) \in L$, there are at most n descendants of u and v. Lines 4-5 take $O(n)$ time since the list $L(I)$ also has size at most n. Defining u_{Smax} and v_{Tmax} and then checking the conditions in lines 7 and 9 also take $O(n)$ time since $L(I)_T$ and $L(I)_S$ have size at most n.

Now in Step 2 of all three cases, ordering the lists $L(I)_T$ and $L(I)_S$ can be done in $O(n \log n)$ time. Next, each of the E sets can be calculated in $O(n)$ time and space, and we perform these calculations at most n times for a total of $O(n^2)$ time and space. The paired flip and subtree switch choices in Step 3 of Algorithm 3, 8, or 4 run in $O(n)$ time each since counting crossings involving $(t_i, s_{\phi(i)})$ can be done in linear time, and performing paired flips and subtree switches can also be done in linear time. We make at most n paired flip and subtree switch choices for a total of $O(n^2)$ time. In Algorithm 4, we may also consider four layouts corresponding to combinations of $\{u_0, v_0\}$ in lines 28-29. Since each layout takes $O(n^2)$ space and crossings can be counted in $O(n^2)$ time by checking all $\binom{n}{2}$ pairs of edges, these remaining steps also take $O(n^2)$ time and space. Thus, the `Insertion` algorithm runs in $O(n^2)$ time and space, regardless of the case. □

If there exists a solution to the Tanglegram Layout Problem for (T, S, ϕ) such that all crossings involve a single edge, then `Insertion` can be used to find a solution to the Tanglegram Layout Problem. In particular, one can use `Insertion`(T, S, ϕ, i) over all possible i to verify if $\operatorname{crt}(T, S, \phi) = 1$. However, in general, the solution to the Single Edge Insertion Problem and the Layout Problem can differ by arbitrarily many crossings.

Definition 39. Let (T, S, ϕ) be a tanglegram of size n. For any $i \in [n]$ such that $(T_I, S_{\phi(I)}, \phi|_I)$ with $I = [n] \setminus \{i\}$ is planar, define $\mathrm{crtei}((T, S, \phi), i)$ to be the common number of crossings in any solution to the Tanglegram Single Edge Insertion Problem. Define

$$\mathrm{crtei}(T, S, \phi) = \min_i\{\mathrm{crtei}((T, S, \phi), i)\}, \tag{8}$$

where the minimum is taken over all well-defined choices of i.

Corollary 40. *For any $k \in \mathbb{N}$, there exists a tanglegram (T, S, ϕ) and such that*

$$\mathrm{crtei}(T, S, \phi) - \mathrm{crt}(T, S, \phi) = k.$$

Proof. For $k = 1$, consider the tanglegram (T, S, ϕ) with two layouts in Figure 18. Direct application of `ModifiedUntangle` or the Tanglegram Kuratowski Theorem in [5] shows that the induced subtanglegram on $[n] \setminus \{i\}$ is planar only when $i = 2$, and hence $\mathrm{crtei}(T, S, \phi) = \mathrm{crtei}((T, S, \phi), 2)$. `Insertion`$((T, S, \phi), 2)$ outputs the first layout in Figure 18 with three crossings, which is a solution to the Tanglegram Single Edge Insertion Problem by Theorem 3. Hence, $\mathrm{crtei}(T, S, \phi) = 3$. Theorem 3 also implies that if $\mathrm{crt}(T, S, \phi) = 1$, then for some choice of $i \in [6]$, `Insertion`$((T, S, \phi), i)$ will have one crossing. Since this not the case, the second layout in Figure 18 implies $\mathrm{crt}(T, S, \phi) = 2$. Combined, we see that $\mathrm{crtei}(T, S, \phi) - \mathrm{crt}(T, S, \phi) = 1$.

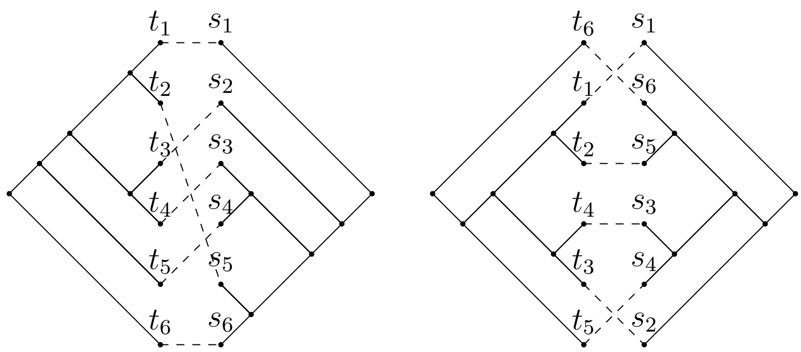

Figure 18: The output of `Insertion`$((T, S, \phi), 2)$ *and another layout for the same tanglegram.*

For $k > 1$, form the size $k + 5$ tanglegram (T', S', ϕ') by starting with (T, S, ϕ) and replacing the edge (t_4, s_3) with a planar tanglegram $(\widetilde{T}, \widetilde{S}, \widetilde{\phi})$ of size k. Since (T', S', ϕ') contains (T, S, ϕ) as a subtanglegram, we know $\mathrm{crt}(T', S', \phi') \geqslant 2$, as any layout of (T', S', ϕ') with fewer than two crossings would restrict to a layout of (T, S, ϕ) with fewer than two crossings. We can take the second layout in Figure 18 and replace (t_4, s_3) with a planar layout of $(\widetilde{T}, \widetilde{S}, \widetilde{\phi})$ to obtain a layout of (T', S', ϕ') with exactly 2 crossings, so $\mathrm{crt}(T', S', \phi') = 2$.

Since $(T_{[6]\setminus\{i\}}, S_{\phi([6]\setminus\{i\})}, \phi|_{[6]\setminus\{i\}})$ is planar only when $i = 2$, the same statement is true for $(T'_{[k+5]\setminus\{i\}}, S'_{\phi([k+5]\setminus\{i\})}, \phi'|_{[k+5]\setminus\{i\}})$, as the tanglegram $(T'_{[k+5]\setminus\{i\}}, S'_{\phi([k+5]\setminus\{i\})}, \phi'|_{[k+5]\setminus\{i\}})$

contains $(T_{[6]\setminus\{i\}}, S_{\phi([6]\setminus\{i\})}, \phi|_{[6]\setminus\{i\}})$ as an induced subtanglegram. Starting with the first layout in Figure 18, construct the layout (X', Y') for (T', S', ϕ') by replacing (t_4, s_3) with a planar layout of $(\widetilde{T}, \widetilde{S}, \widetilde{\phi})$. Notice that (X', Y') has exactly $k+2$ crossings and restricts to a planar layout of $(T'_{[k+5]\setminus\{2\}}, S'_{\phi([k+5]\setminus\{2\})}, \phi'|_{[k+5]\setminus\{2\}})$. Furthermore, notice that the only leaf-matched pairs of $(T'_{[k+5]\setminus\{2\}}, S'_{\phi([k+5]\setminus\{2\})}, \phi'|_{[k+5]\setminus\{2\}})$ are the roots of the two trees and leaf-matched pairs of $(\widetilde{T}, \widetilde{S}, \widetilde{\phi})$, so Corollary 32 implies that a solution to the Tanglegram Single Edge Insertion Problem for (T', S', ϕ') and $i = 2$ can be obtained by starting with (X', Y') and using subtree switches at the parents of t_2 and s_5. However, these subtree switches increase the number of crossings, so (X', Y') itself must be a solution to the Single Edge Insertion Problem. Hence, $\text{crtei}(T', S', \phi') = \text{crtei}((T', S', \phi'), 2) = k + 2$, and therefore $\text{crtei}(T', S', \phi') - \text{crt}(T', S', \phi') = k$. □

5 Multiple edge insertion

In this section, we consider the Tanglegram Multiple Edge Insertion Problem, which we restate below for convenience of the reader. Similar to the corresponding problem for graphs, this problem is NP-hard, which we now show.

Problem (**Tanglegram Multiple Edge Insertion**). Given a tanglegram (T, S, ϕ) and a planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ induced by $I \subseteq [n]$, find a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ and has the minimal number of crossings possible.

Proof of Theorem 4. By Theorem 11, the crossing number of any tanglegram (T, S, ϕ) of size n is strictly less than $\frac{1}{2}\binom{n}{2}$. Then in an optimal layout of a tanglegram (T, S, ϕ), there must exist some edges $(t_i, s_{\phi(i)})$ and $(t_j, s_{\phi(j)})$ that do not cross. If we solve the Multiple Edge Insertion Problem for all index sets I of size two, one of them will be $I = \{i, j\}$. After all $\binom{n}{2} = O(n^2)$ iterations, one of these layouts will be a solution to the Tanglegram Layout Problem, which is NP-hard by Theorem 9. □

5.1 Iterated single edge insertion

We start by giving an algorithm that inserts a single edge at a time using the approach from the `Insertion` algorithm. While this will not solve the Tanglegram Multiple Edge Insertion Problem, it has two advantages. First, the algorithm will run in polynomial time. Second, Theorem 10 implies that if $(T_I, S|_{\phi(I)}, \phi|_I)$ is a size $n-1$ subtanglegram of the size n tanglegram (T, S, ϕ), then $\text{crt}(T, S, \phi)$ and $\text{crt}(T_I, S|_{\phi(I)}, \phi|_I)$ differ by at most $n-3$. This can be used to obtain a general bound on a tanglegram's crossing number as a function of the number of edges that we remove to obtain a planar subtanglegram. The algorithm we give in this subsection will achieve this bound.

Lemma 41. *Let (T, S, ϕ) be a size n tanglegram, and let $I \subseteq [n]$. If $(T_I, S_{\phi(I)}, \phi|_I)$ is planar, then* $\text{crt}(T, S, \phi) \leqslant \frac{(n-|I|)\cdot(n+|I|-5)}{2}$.

Proof. We assume without loss of generality that $I = \{1, 2, \ldots, k\}$, as otherwise we can relabel the tanglegram. Using the assumption $\operatorname{crt}(T_I, S_{\phi(I)}, \phi|_I) = 0$ with Theorem 10,

$$\begin{aligned}\operatorname{crt}(T, S, \phi) &= \sum_{j=k+1}^{n} \left[\operatorname{crt}(T_{[j]}, S_{\phi([j])}, \phi|_{[j]}) - \operatorname{crt}(T_{[j-1]}, S_{\phi([j-1])}, \phi|_{[j-1]})\right] \\ &\leqslant \sum_{j=k+1}^{n} (j-3) = \frac{(n-k)(n+k-5)}{2}. \end{aligned}$$

□

We now give an algorithm that starts with a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ and inserts one edge at a time, with operations chosen in a similar manner as in `Insertion`. As before, we restrict ourselves to paired flips and subtree switches, though we perform a paired flip at each $(u, v) \in L(I)$ at most once to ensure we achieve the bound in Lemma 41.

Algorithm 6: `IteratedInsertion`

Input: tanglegram (T, S, ϕ), index set I such that $(T_I, S_{\phi(I)}, \phi|_I)$ is planar
Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

1 $(X, Y), L(I) \coloneqq$ `ModifiedUntangle`$(T_I, S_{\phi(I)}, \phi|_I)$ $J \coloneqq I$, $M \coloneqq L(I)$
2 **while** $[n] \setminus J \neq \emptyset$, **do**
3     $i \coloneqq$ smallest element of $[n] \setminus J$ `// Choose an edge to insert.`
4   extend (X, Y) to a layout of $(T_{J\cup\{i\}}, S_{\phi(J\cup\{i\})}, \phi|_{J\cup\{i\}})$
5     $M_T \coloneqq \{(u, v) \in M : u >_T t_i, v \not>_S s_{\phi(i)}\}$ `// Perform paired flips at`
6     $M_S \coloneqq \{(u, v) \in M : u \not>_T t_i, v >_S s_{\phi(i)}\}$ `// these pairs of vertices.`
7   update $M \coloneqq M \setminus (M_T \cup M_S)$ `// Prevent future flips in` $M_T \cup M_S$.
8     $u_0, v_0 \coloneqq$ parents of $t_i, s_{\phi(i)}$ in $(T_{J\cup\{i\}}, S_{\phi(J\cup\{i\})}, \phi|_{J\cup\{i\}})$
9   proceed from Step 2 of `Insertion` with $L(I)_T, L(I)_S$ respectively replaced by M_T, M_S, and obtain (X, Y) after operations at M_T, M_S, u_0, v_0
10   update $J \coloneqq J \cup \{i\}$
11 **return** (X, Y)

Theorem 42. *Let (T, S, ϕ) be a size n tanglegram and $I \subseteq [n]$ such that $(T_I, S_{\phi(I)}, \phi|_I)$ is planar. `IteratedInsertion` finds a layout of (T, S, ϕ) with at most $\frac{(n-|I|)\cdot(n+|I|-5)}{2}$ crossings in $O(n^3)$ time and $O(n^2)$ space.*

Proof. We start by proving the run-time and space claims. Line 1 runs in $O(n^2)$ time and space. By construction, the **while** loop will have at most $n - |I|$ iterations since we add an element to J at the end of each iteration. Line 4 can be done in $O(n^2)$ time and space by starting with the trees $T_{J\cup\{i\}}$ and $S_{\phi(J\cup\{i\})}$ and performing operations until the leaves of T_J and $S_{\phi(J)}$ appear in the order indicated by (X, Y). The remaining steps of the **while** loop also run in $O(n^2)$ time since `Insertion` runs in $O(n^2)$ time, for a combined total of $O(n^3)$ time. Since we re-use the same variables X, Y, J, M, M_T, M_S, we also see that the algorithm runs in $O(n^2)$ space since `Insertion` runs in $O(n^2)$ space by Theorem 3.

We now show that the output has at most $\frac{(n-|I|)\cdot(n+|I|-5)}{2}$ crossings. First, note that since the subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ is planar, the layout (X, Y) originally obtained from `ModifiedUntangle`$(T_I, S_{\phi(I)}, \phi|_I)$ in line 1 has 0 crossings. We show that during each iteration of the **while** loop, the algorithm adds at most $|J| - 3$ crossings. This would imply that the total number of crossings at the end of `IteratedInsertion` is bounded by the sum from the proof of Lemma 41, implying the claim.

We consider two cases for each iteration of the **while** loop. First, suppose that in this iteration of the loop, the sets M_T and M_S are empty. Then the algorithm only considers subtree switches at u_0 and v_0, the parents of t_i and $s_{\phi(i)}$. In particular, we do not affect the relative ordering of any vertices in $(T_J, S_{\phi(J)}, \phi|_J)$, and thus we do not affect any previous crossings. Some combination of subtree switches will produce at most $|J|-3$ of the $|J|-1$ possible crossings. Proceeding from Step 2 of `Insertion` will lead to using Algorithm 4, which will consider subtree switches at u_0 and v_0 before returning the option with the fewest number of crossings. Thus, the algorithm adds at most $|J| - 3$ new crossings from this iteration, all involving edge $(t_i, s_{\phi(i)})$.

Next, suppose that $M_T \cup M_S$ is nonempty, and let $(u, v) \in M_T \cup M_S$. By Lemma 28, (u, v) is a leaf-matched pair of $(T_I, S_{\phi(I)}, \phi|_I)$. Since (u, v) was not previously considered in the algorithm, it must be that for all previously inserted edges (t, s), both or neither of $u >_T t$ and $v >_S s$ are true. From this, we see that crossings between (t, s) and edges in $(T_I, S_{\phi(I)}, \phi|_I)$ are not affected by paired flips at (u, v). Furthermore, for any other previously inserted edge (t', s'), we also know that either both or neither of $u >_T t'$ and $v >_S s'$ are true. In all cases, whether or not (t, s) and (t', s') intersect is not affected by a paired flip at (u, v). Thus, all previously existing crossings are unaffected, and we only create new crossings involving the current edge $(t_i, s_{\phi(i)})$. Using the same argument as in the previous paragraph, proceeding from Step 2 of `Insertion` will add at most $|J| - 3$ new crossings, all of which involve edge $(t_i, s_{\phi(i)})$. □

5.2 MultiInsertion Algorithm

We now generalize our results from Section 4 and then give an algorithm for solving the Tanglegram Multiple Edge Insertion Problem. For this subsection, fix a tanglegram (T, S, ϕ) of size n, and fix $I \subseteq [n]$ such that $(T_I, S_{\phi(I)}, \phi|_I)$ is a planar subtanglegram. Let (X, Y) be a layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$, and let $L(I)$ be the set of leaf-matched pairs of $(T_I, S_{\phi(I)}, \phi|_I)$. Recall that for $u \in T$ and $v \in S$, $\mathrm{Lf}(u)$ and $\mathrm{Lf}(v)$ respectively denote the descendants of u and v that are leaves in T and S. For $J \subseteq [n]$, we will abuse notation by using $\mathrm{Lf}(u) \cap J$ or $\mathrm{Lf}(v) \cap \phi(J)$ to denote the leaves in $\mathrm{Lf}(u)$ or $\mathrm{Lf}(v)$ that are indexed by J or $\phi(J)$, respectively. Letting I^c denote the complement of I, define the following sets:

$$\begin{aligned}
L(I)_0 &= \{(u, v) \in L(I) : \mathrm{Lf}(u) \cap I^c \text{ is matched with } \mathrm{Lf}(v) \cap \phi(I)^c \text{ by } \phi\} \\
L(I)_T &= \{(u, v) \in L(I) : \ |\mathrm{Lf}(u) \cap I^c| = 1 \text{ and } |\mathrm{Lf}(v) \cap \phi(I)^c| = 0\} \\
L(I)_S &= \{(u, v) \in L(I) : \ |\mathrm{Lf}(u) \cap I^c| = 0 \text{ and } |\mathrm{Lf}(v) \cap \phi(I)^c| = 1\} \\
L(I)_1 &= L(I) \setminus (L(I)_0 \cup L(I)_T \cup L(I)_S) \\
M(I) &= \{\text{internal vertices of } (T, S, \phi) \text{ that are not in } (T_I, S_{\phi(I)}, \phi|_I).\}
\end{aligned} \tag{9}$$

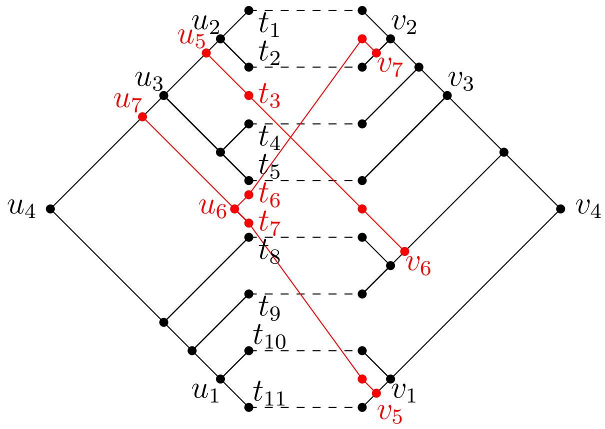

Figure 19: The subtanglegram induced by $I = \{1, 2, 4, 5, 8, 9, 10, 11\} \subseteq [11]$ is shown in black. By (9), we see that $L(I) = \{(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4)\}$ and $M(I) = \{u_5, u_6, u_7, v_5, v_6, v_7\}$. Using (9), we partition $L(I)$ into $L(I)_0 = \{(u_4, v_4)\}$, $L(I)_T = \emptyset$, $L(I)_S = \{(u_1, v_1), (u_2, v_2)\}$, and $L(I)_1 = \{(u_3, v_3)\}$.

Notice that $L(I)_0, L(I)_T, L(I)_S$, and $L(I)_1$ partition the set $L(I)$. When $|I| = n - 1$, the sets $L(I)_0$, $L(I)_T$ and $L(I)_S$ reduce to the same ones defined in Section 4, and $L(I)_1 = \emptyset$. An example of these sets is shown in Figure 19.

Lemma 43. *Let (X', Y') be the image of (X, Y) after any sequence of paired flips at elements in $L(I)$ and subtree switches at elements of $M(I)$. Then (X', Y') also restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$.*

Proof. It suffices to show the claim when (X', Y') is the image of (X, Y) after a single subtree switch or paired flip. If we start with (X, Y) and perform a flip at $(u, v) \in L(I)$, then restricted to the subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$, this has the same effect as performing a paired flip on a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$. Thus, (X', Y') is still planar when restricted to $(T_I, S_{\phi(I)}, \phi|_I)$.

Now suppose we start with (X, Y) and perform a subtree switch at $u \in M(I)$ to obtain (X', Y'). Without loss of generality, we assume $u \in T$, as $u \in S$ is done similarly. Then we can decompose $(X, Y) = (X_1U_1U_2X_2, Y)$ and $(X', Y') = (X_1U_2U_1X_2, Y)$, where U_1 and U_2 order the leaves of the subtrees rooted at the children of u. Since u is not a vertex of $(T_I, S_{\phi(I)}, \phi|_I)$, Definitions 6 and 7 imply that either

(1) u is not in the minimal subtree of T containing the leaves indexed by I, or

(2) u is one of the internal vertices with one child that is suppressed when forming T_I from T.

In the case of (1), the entire subtree rooted at u does not appear in T_I. Then both U_1 and U_2 are deleted when we restrict (X, Y) and (X', Y') to $(T_I, S_{\phi(I)}, \phi|_I)$, and both layouts restrict to the same layout of the subtanglegram. In the case of (2), the entire subtree rooted at one of the children of u does not appear in T_I, so either U_1 or U_2 is deleted when (X, Y) and (X', Y') are restricted to $(T_I, S_{\phi(I)}, \phi|_I)$. Again, (X', Y') and (X, Y) will restrict to the exact same layout of $(T_I, S_{\phi(I)}, \phi|_I)$ as (X, Y). In both cases, since (X, Y) restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$, so does (X', Y'). □

Lemma 44. *Let (X', Y') be the image of (X, Y) after a paired flip at $(u, v) \in L(I)_0$. Then (X, Y) and (X', Y') have the same crossings.*

Proof. Since $(u, v) \in L(I)_0$, notice that both or neither of $u >_T t_i$ and $v >_S s_{\phi(i)}$ are true for any $i \notin I$. If $u >_T t_i$ and $v >_S s_{\phi(i)}$, a paired flip at (u, v) flips the induced subtanglegram on the subtrees rooted at u and v, preserving all crossings in (T, S, ϕ) that involve $(t_i, s_{\phi(i)})$. Otherwise, if $u \not>_T t_i$ and $v \not>_S s_{\phi(i)}$, then $(t_i, s_{\phi(i)})$ cross all or none of the edges between the subtrees rooted at u and v, and a paired at (u, v) does not affect these crossings. □

Lemma 45. *A solution to the Tanglegram Multiple Edge Insertion Problem can be obtained by starting at (X, Y) and performing a sequence of subtree switches at elements in $M(I)$ and paired flips at elements in $L(I)_T \cup L(I)_S \cup L(I)_1$.*

Proof. Let (X_{min}, Y_{min}) be a solution to the Multiple Edge Insertion Problem. Some composition of flips $f_m \circ \ldots \circ f_2 \circ f_1$ maps (X, Y) to (X_{min}, Y_{min}), as flips generate all trees isomorphic to T and S. All of these flips commute and have order 2, so we can also assume that all f_i are distinct, i.e., no flips occur at any vertex more than once.

If none of the flips in $\{f_1, f_2, \ldots, f_m\}$ involve vertices in $M(I)$, then restricting (X, Y) and (X_{min}, Y_{min}) to the subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$, these flips are equivalent to a sequence of flips mapping one planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$ to another. By Theorem 1, these flips must be equivalent to a sequence of paired flips at elements in $L(I)$. Since all flips commute, we can commute paired flips involving $(u, v) \in L(I)_0$ to be the last ones performed. By Lemma 44, paired flips at $(u, v) \in L(I)_0$ do not affect any crossings, so we can obtain another solution to the Multiple Edge Insertion Problem by excluding them, leaving only the paired flips at elements in $L(I) \setminus L(I)_0 = L(I)_T \cup L(I)_S \cup L(I)_1$.

Now suppose some flip in $\{f_1, f_2, \ldots, f_m\}$ involves an element in $M(I)$. Without loss of generality, assume f_1 is a flip at $u \in M(I)$, $u \in T$, and no flips occur at any $u' >_T u$. Now define g_1 and h_1 to be flips at the children of u, or treat these as the identity if the corresponding child is a leaf. Notice that the composition

$$f_m \circ \ldots \circ f_2 \circ h_1 \circ g_1 \circ h_1 \circ g_1 \circ f_1$$

also maps (X, Y) to (X_{min}, Y_{min}). The composition $h_1 \circ g_1 \circ f_1$ is a subtree switch at u, and the remaining operations $g_1, h_1, f_2, \ldots, f_m$ do not involve flips at any $u' \in M(I)$ with $u' \geqslant_T u$. If g_1 or h_1 are equivalent to any of the f_i, then we commute flips and replace $f_i \circ g_1$ or $f_i \circ h_1$ with the identity. We will find that $f_m \circ \cdots \circ f_2 \circ h_1 \circ g_1$ is equivalent to $\widetilde{f}_k \circ \cdots \circ \widetilde{f}_2 \circ \widetilde{f}_1$, and in this second composition, there are no repeated flips at any vertex, and no flips occur at $u' \in M(I)$ with $u' \geqslant_T u$. If some $\widetilde{f}_i$ involves a flip at a vertex in $M(I)$, then we iterate this argument. Since we choose a maximal vertex $u \in M(I)$ at each iteration, this process will eventually terminate. We will then have a sequence of subtree switches mapping (X, Y) to some layout (X', Y'), and the remaining flips mapping (X', Y') to (X_{min}, Y_{min}) do not involve vertices in $M(I)$. The conclusion then follows from the preceding paragraph. □

We now focus on $M(I)$ and define subsets $M(I)_0$, $M(I)_T$, and $M(I)_S$ that allow us to generalize our results from `Insertion`. For any leaves t_i and $s_{\phi(i)}$ with $i \notin I$, we let $P(t_i)$ and $P(s_{\phi(i)})$ denote their respective parents. We define $M(I)_0$ to contain all $P(t_i)$ and $P(s_{\phi(i)})$ for $i \notin I$, such that

- $\mathrm{Lf}(P(t_i)) \cap I^c = \{t_i\}$,
- $\mathrm{Lf}(P(s_{\phi(i)})) \cap \phi(I)^c = \{s_{\phi(i)}\}$, and
- there exists $(u, v) \in L(I)$ with either $P(t_i) >_T u$ and $v >_S P(s_{\phi(i)})$, or $P(s_{\phi(i)}) >_S v$ and $u >_T P(t_i)$.

Note that these combined properties imply $(u, v) \notin L(I)_0$ by definition. The set $M(I)_0$ is intended to generalize the results in Algorithms 3 and 8, which are the cases $u_0 >_T u_{Smax}$ and $v_0 >_S v_{Tmax}$.

Now we define $M(I)_T$ and $M(I)_S$, which generalize additional situations from our `Insertion` algorithm. For any $t_i \in T$ with $i \notin I$, define $A(t_i)$ to be the minimal distance ancestor u of t_i such that $\mathrm{Lf}(u) \cap I \neq \emptyset$. Now define $M(I)_S$ to contain all $P(s_{\phi(i)}) \in M(I) \setminus M(I)_0$ for $i \notin I$ such that $\mathrm{Lf}(P(s_{\phi(i)})) \cap \phi(I)^c = \{s_{\phi(i)}\}$, and either

- $\mathrm{Lf}(A(t_i)) \cap I$ and $\mathrm{Lf}(P(s_{\phi(i)})) \cap \phi(I)$ do not contain matched leaves, or
- there exists $(u, v) \in L(I)$ such that $A(t_i) >_T u$ and $v >_S P(s_{\phi(i)})$. In this case, note that these properties imply $(u, v) \notin L(I)_0$.

Similarly, define $A(s_{\phi(i)})$ to be the minimal distance ancestor v of $s_{\phi(i)}$ such that $\mathrm{Lf}(v) \cap \phi(I) \neq \emptyset$. Define $M(I)_T$ to contain all $P(t_i) \in M(I) \setminus M(I)_0$ for $i \notin I$ such that $\mathrm{Lf}(P(t_i)) \cap I^c = \{t_i\}$, and either

- $\mathrm{Lf}(P(t_i)) \cap I$ and $\mathrm{Lf}(A(s_{\phi(i)})) \cap \phi(I)$ do not contain matched leaves, or
- there exists $(u, v) \in L(I)$ such that $u >_T P(t_i)$ and $A(s_{\phi(i)}) >_S v$. Again, $(u, v) \notin L(I)_0$.

Finally, we define $M(I)_1 = M(I) \setminus (M(I)_0 \cup M(I)_T \cup M(I)_S)$ to be the remaining inserted vertices so that $M(I)_0, M(I)_T, M(I)_S$, and $M(I)_1$ partition $M(I)$. An example of these sets is shown in Figure 20.

We now define a natural partial order $\preceq$ on $\mathcal{P} = L(I)_T \cup L(I)_S \cup L(I)_1 \cup M(I)$ using the partial orders on T and S. This will be useful for both determining what crossings to focus on at each element, as well as the order in which we should consider the elements when performing subtree switches and paired flips.

- For $(u_1, v_1), (u_2, v_2) \in \mathcal{P} \cap L(I)$, $(u_1, v_1) \preceq (u_2, v_2)$ if $u_1 \leqslant_T u_2$ and $v_1 \leqslant_S v_2$.

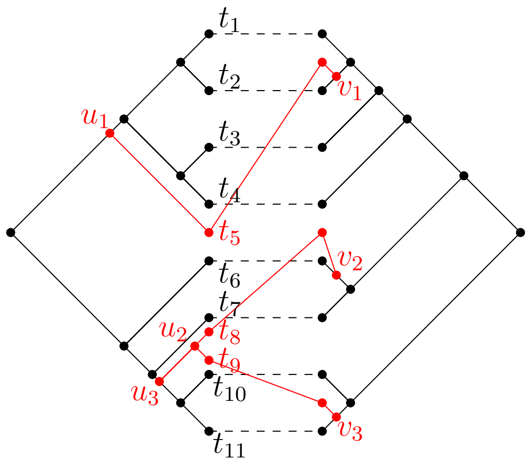

Figure 20: For the subtanglegram induced by $I = \{1, 2, 3, 4, 6, 7, 10, 11\} \subseteq [11]$ shown in black, $M(I)$ is partitioned into $M(I)_0 = \{u_1, v_1\}$, $M(I)_T = \emptyset$, $M(I)_S = \{v_2, v_3\}$, and $M(I)_1 = \{u_2, u_3\}$.

- For $(u_1, v_1) \in \mathcal{P} \cap L(I), u \in \mathcal{P} \cap T$, $(u_1, v_1) \preceq u$ if $u_1 \leqslant_T u$, and $u \preceq (u_1, v_1)$ if $u \leqslant_T u_1$.
- For $(u_1, v_1) \in \mathcal{P} \cap L(I), v \in \mathcal{P} \cap S$, $(u_1, v_1) \preceq v$ if $v_1 \leqslant_S v$, and $v \preceq (u_1, v_1)$ if $v \leqslant_S v_1$.
- For $u_1, u_2 \in \mathcal{P} \cap T$, $u_1 \preceq u_2$ if $u_1 \leqslant_T u_2$.
- For $v_1, v_2 \in \mathcal{P} \cap S$, $v_1 \preceq v_2$ if $v_1 \leqslant_S v_2$.
- Finally, take the transitive closure of the above relations.

Recall that for partially ordered sets, we say $z \in \mathcal{P}$ *covers* $x \in \mathcal{P}$ if $x \prec z$ and there do not exist any $y \in \mathcal{P}$ with $x \prec y \prec z$, i.e., z is larger than x and there are no elements between them. We use the notation $x \succ\!\cdot\, y$ or $y \prec\!\cdot\, x$ to denote the covering relation.

For the elements in $L(I)_T, L(I)_S, M(I)_0, M(I)_T$, and $M(I)_S$, we now define generalizations of the $E(u_j)$ and $E(v_j)$ sets from `Insertion`. These sets, which we denote $\mathrm{Crs}(u, v)$, $\mathrm{Crs}(u)$, or $\mathrm{Crs}(v)$, will determine what crossings we should consider at the corresponding elements. Using the notation e_j for the between-tree edge $(t_j, s_{\phi(j)})$, we organize these definitions in Table 2.

Finally, recall that a *linear extension* of a partially ordered set $\mathcal{P}$ is a bijective map $F : \mathcal{P} \to \{1, 2, \ldots, |\mathcal{P}|\}$ such that $F(w) \geqslant F(w')$ in the usual order on $\mathbb{N}$ whenever $w \succeq w'$, or equivalently, a total ordering of $\mathcal{P}$ that respects the partial ordering. We can use a linear extension on $\mathcal{P}$ with respect to $\preceq$ defined above to determine the order that we perform operations, as by construction of $\preceq$, the order $F^{-1}(|\mathcal{P}|), \ldots, F^{-1}(2), F^{-1}(1)$ guarantees that we consider all ancestors of a vertex in T or S before we consider the vertex itself. We now give an example of the sets in Table 2, followed by our `MultiInsertion` algorithm and an example of its application.

Element	Relevant Properties	Crossing Set
$(u,v) \in L(I)_T$	$\mathrm{Lf}(u) \cap I^c = \{t_i\}$	$\{(e_i,e_j) : t_j \in \mathrm{Lf}(u) \setminus \mathrm{Lf}(u')$ where $(u,v) \succ u'$ or $(u,v) \succ (u',v')\}$
$(u,v) \in L(I)_S$	$\mathrm{Lf}(v) \cap \phi(I)^c = \{s_{\phi(i)}\}$	$\{(e_i,e_j) : s_{\phi(j)} \in \mathrm{Lf}(v) \setminus \mathrm{Lf}(v')$ where $(u,v) \succ v'$ or $(u,v) \succ (u',v')\}$
$P(t_i) \in M(I)_0$	there exists $(u,v) \in L(I)$ with $P(t_i) >_T u$ and $v >_S P(s_{\phi(i)})$	$\{(e_i,e_j) : t_j \in \mathrm{Lf}(P(t_i)) \setminus \mathrm{Lf}(u)$ where $P(t_i) \succ (u,v)\}$
$P(s_{\phi(i)}) \in M(I)_0$	there exists $(u,v) \in L(I)$ with $P(t_i) >_T u$ and $v >_S P(s_{\phi(i)})$	$\{(e_i,e_j) : s_{\phi(j)} \in \mathrm{Lf}(P(s_{\phi(i)})) \setminus \{s_{\phi(i)}\}\}$
$P(t_i) \in M(I)_0$	there exists $(u,v) \in L(I)$ with $P(s_{\phi(i)}) >_S v$ and $u >_T P(t_i)$	$\{(e_i,e_j) : t_j \in \mathrm{Lf}(P(t_i)) \setminus \{t_i\}\}$
$P(s_{\phi(i)}) \in M(I)_0$	there exists $(u,v) \in L(I)$ with $P(s_{\phi(i)}) >_S v$ and $u >_T P(t_i)$	$\{(e_i,e_j) : s_{\phi(j)} \in \mathrm{Lf}(P(s_{\phi(i)})) \setminus \mathrm{Lf}(v)$ where $P(s_{\phi(i)}) \succ (u,v)\}$
$P(t_i) \in M(I)_T$	$\mathrm{Lf}(P(t_i)) \cap I^c = \{t_i\}$	$\{(e_i,e_j) : t_j \in \mathrm{Lf}(P(t_i)) \setminus \{t_i\}\}$
$P(s_{\phi(i)}) \in M(I)_S$	$\mathrm{Lf}(P(s_{\phi(i)})) \cap \phi(I)^c = \{s_{\phi(i)}\}$	$\{(e_i,e_j) : t_j \in \mathrm{Lf}(P(s_{\phi(i)})) \setminus \{s_{\phi(i)}\}\}$

Table 2: Crossing sets for elements in $L(I)_T$, $L(I)_S$, $M(I)_0$, $M(I)_T$, and $M(I)_S$.

Example 46. Consider the tanglegram in Figure 21, and let $I = \{1, 2, 4, 5, 8, 9, 10, 11\}$. We index elements in $\mathcal{P}$ by their images under a linear extension, so

$$\mathcal{P} = \{v_1, (u_2, v_2), v_3, v_4, (u_5, v_5), u_6, (u_7, v_7), u_8, u_9\}. \tag{10}$$

Observe that $L(I)_T = \emptyset$, $L(I)_S = \{(u_2, v_2), (u_5, v_5)\}$, $L(I)_1 = \{(u_7, v_7)\}$, $M(I)_0 = \emptyset$, $M(I)_T = \{u_6\}$, $M(I)_S = \{v_1, v_3, v_4\}$, and $M(I)_1 = \{u_8, u_9\}$. Table 2 states $\mathrm{Crs}(v_1) = \{(e_7, e_{11})\}$, $\mathrm{Crs}(u_2, v_2) = \{(e_7, e_{10})\}$, $\mathrm{Crs}(v_3) = \{(e_3, e_8), (e_3, e_9)\}$, $\mathrm{Crs}(v_4) = \{(e_6, e_2)\}$, $\mathrm{Crs}(u_5, v_5) = \{(e_6, e_1)\}$, and $\mathrm{Crs}(u_6) = \{(e_3, e_1), (e_3, e_2)\}$.

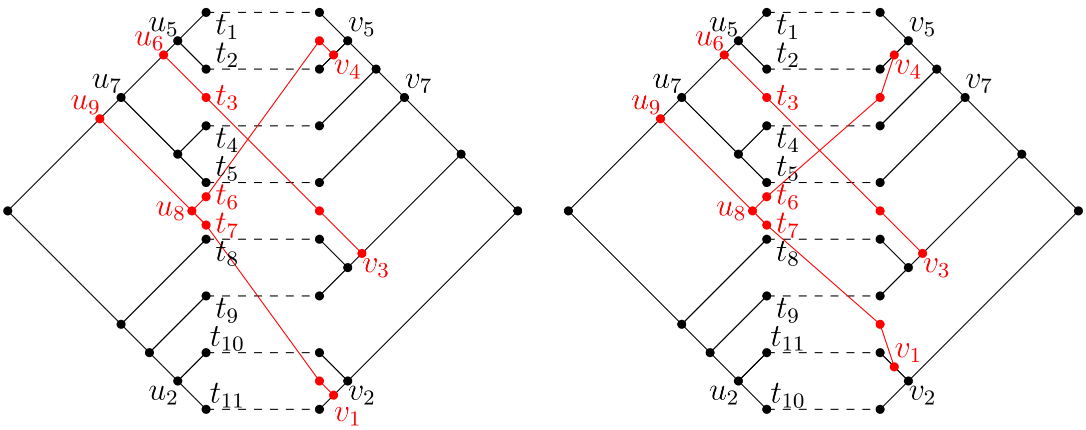

Figure 21: Two layouts for the same tanglegram.

Algorithm 7: `MultiInsertion`

Input: tanglegram (T, S, ϕ), index set I such that $(T_I, S_{\phi(I)}, \phi|_I)$ is planar
Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

1 $(X, Y), L :=$ `ModifiedUntangle`$(T, S, \phi|_I)$
2 construct $L(I)$ from L using Definition 26
3 construct $M(I)$, $M(I)_0$, $M(I)_T$, $M(I)_S$, $M(I)_1$, $L(I)_0$, $L(I)_T$, $L(I)_S$, and $L(I)_1$
4 calculate $\mathrm{Crs}(w)$ for all $w \in L(I)_T \cup L(I)_S \cup M(I)_0 \cup M(I)_T \cup M(I)_S$ using Table 2
5 $F :=$ linear extension of $\mathcal{P} = L(I)_T \cup L(I)_S \cup L(I)_1 \cup M(I)$ with respect to partial order $\preceq$
6 $(X_{min}, Y_{min}) := (X, Y)$
7 **for** $C \subseteq L(I)_1 \cup M(I)_1$, **do**
8     $(X', Y') := (X, Y)$
9   **for** $j = |\mathcal{P}|, \dots, 2, 1$, **do**
10   **if** *[*$F^{-1}(j) \in C \cap L(I)_1$*] or [*$F^{-1}(j) \in L(I)_T \cup L(I)_S$ *and more than half of the crossings in* $\mathrm{Crs}(F^{-1}(j))$ *occur in* (X', Y')*],* **then**
11   update $(X', Y') :=$ `PairedFlip`$((X', Y'), F^{-1}(j))$
12   **else if** *[*$F^{-1}(j) \in C \cap M(I)_1 \cap T$*] or [*$F^{-1}(j) \in M(I)_T \cup (M(I)_0 \cap T)$ *and more than half of the crossings in* $\mathrm{Crs}(F^{-1}(j))$ *occur in* (X', Y')*],* **then**
13   update $X' :=$ `SubtreeSwitch`$(X', F^{-1}(j))$
14   **else if** *[*$F^{-1}(j) \in C \cap M(I)_1 \cap S$*] or [*$F^{-1}(j) \in M(I)_S \cup (M(I)_0 \cap S)$ *and more than half of the crossings in* $\mathrm{Crs}(F^{-1}(j))$ *occur in* (X', Y')*],* **then**
15   update $Y' :=$ `SubtreeSwitch`$(Y', F^{-1}(j))$
16   **if** (X', Y') *has fewer crossings than* (X_{min}, Y_{min})*,* **then**
17   update $(X_{min}, Y_{min}) := (X', Y')$
18 **return** (X_{min}, Y_{min})

Example 47. Consider the tanglegram (T, S, ϕ) and linear extension of $\mathcal{P}$ from Example 46. Suppose that the the output of `ModifiedUntangle`$(T, S, \phi|_I)$ is the layout on the left in Figure 21. `MultiInsertion` will consider $2^{|L(I)_1 \cup M(I)_1|} = 8$ iterations in the **for** loop in line 7. In each iteration, it produces another layout, and ultimately the algorithm returns the layout encountered that has the fewest number of crossings. In the iteration $C = \emptyset$, the algorithm will perform a subtree switch at v_4, a paired flip at (u_2, v_2), and a subtree switch at v_1, resulting in a layout with seven crossings shown on the right in Figure 21. Note that this is not a solution to the Multiple Edge Insertion Problem since the iteration with $C = \{(u_7, v_7)\}$ will produce a layout with fewer crossings.

Lemma 48. *In some iteration of the **for** loop in line 7 of `MultiInsertion`, the resulting layout* (X', Y') *is a solution to the Tanglegram Multiple Edge Insertion Problem.*

Proof. By Lemma 45, starting with (X, Y) in line 1 and performing some sequence of paired flips and subtree switches at elements in $\mathcal{P}$ produces a solution. In some iteration of the **for** loop in line 7, the choices at elements in $L(I)_1 \cup M(I)_1$ all extend to a solution.

We will assume that we are in this iteration, and we will show that the choice at each element in $L(I)_T \cup L(I)_S \cup M(I)_0 \cup M(I)_T \cup M(I)_S$ extends to a solution, provided that all prior choices extend to a solution.

Consider $(u, v) \in L(I)_T$. By definition of $L(I)_T$, u is an ancestor of a single inserted leaf t_i and v is not an ancestor of any inserted leaves. Thus, the choice at (u, v) can only affect the crossings $C = \{(e_i, e_j) : t_j \in \mathrm{Lf}(u)\}$. Applying the same arguments as in Lemma 34 and 38, if we perform a paired flip at (u, v), then we can perform either a paired flip or a subtree flip at the element covered by (u, v) to obtain the same crossings in $C \setminus \mathrm{Crs}(u, v)$. The case of a paired flip is illustrated in Figure 22. From this, we see that the choice at (u, v) that extends to a solution must be one that minimizes crossings in $\mathrm{Crs}(u, v)$, which is what `MultiInsertion` does. The case $(u, v) \in L(I)_S$ follows by a similar argument.

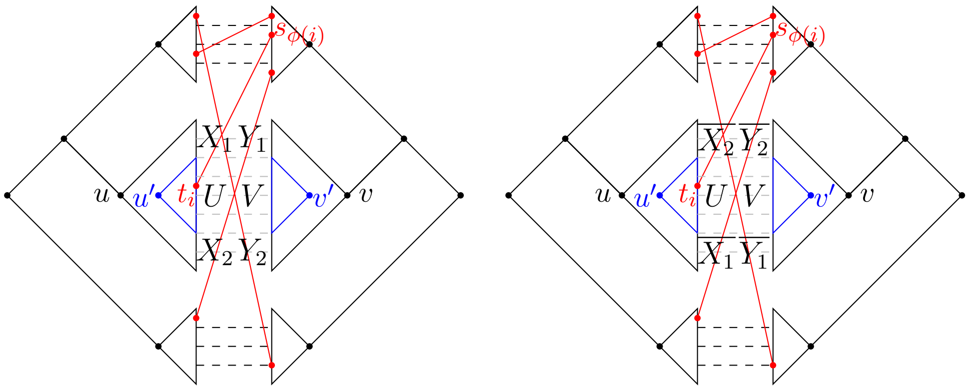

Figure 22: The effect of a paired flip at $(u, v) \in L(I)_T$ and (u', v') with $(u, v) \succ (u', v')$

Now consider $P(t_i), P(s_{\phi(i)}) \in M(I)_0$. Our choices at $P(t_i), P(s_{\phi(i)})$ can only affect crossings between $(t_i, s_{\phi(i)})$ and edges in $(T_I, S_{\phi(I)}, \phi|_I)$ with an endpoint in $\mathrm{Lf}(P(t_i))$ or $\mathrm{Lf}(P(s_{\phi(i)}))$. In the case that there is a leaf-matched pair $(u, v) \in L(I)$ with $P(t_i) >_T u$ and $v >_S P(s_{\phi(i)})$, we can use the same arguments from Lemma 34 with $P(t_i)$ in place of u_0 and $P(s_{\phi(i)})$ in place of v_0 to conclude that `MultiInsertion` makes choices that extend to a solution. The case when there is a leaf-matched pair $(u, v) \in L(I)$ such that $P(s_{\phi(i)}) >_S v$ and $u >_T P(t_i)$ is done similarly.

Finally, consider $P(t_i) \in M(I)_T$. Notice that the choice at $P(t_i)$ can only affect the crossings $\mathrm{Crs}(P(t_i)) = \{(e_i, e_j) : t_j \in \mathrm{Lf}(P(t_i)) \setminus \{t_i\}\}$, where all such t_j must be leaves in the subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$ since $\mathrm{Lf}(P(t_i)) \cap I^c = \{t_i\}$. We claim that our future choices at elements in $\mathcal{P}$ do not affect these crossings.

- Any leaf-matched pairs $(u, v) \in L(I)$ with $(u, v) \succ P(t_i)$ have already been considered based on the linear extension, and there are no leaf-matched pairs $(u, v) \in L(I)$ with $P(t_i) \succ (u, v)$.
- If we perform subtree switches at $u \in M(I) \cap T \setminus \{P(t_i)\}$, then the resulting layouts restrict to the same layout of $(T_{I\cup\{i\}}, S_{\phi(I\cup\{i\})}, \phi|_{I\cup\{i\}})$. Hence, the same crossings occur in $\mathrm{Crs}(P(t_i))$.

- If we perform subtree switches at $v \in M(I) \cap S$ with $v \neq A(s_{\phi(i)})$, then the resulting layouts restrict to the same layout of $(T_{I\cup\{i\}}, S_{\phi(I\cup\{i\})}, \phi|_{I\cup\{i\}})$. Again, the same crossings occur in $\mathrm{Crs}(P(t_i))$.
- Now for $A(s_{\phi(i)})$, we consider two cases. In the case that $\mathrm{Lf}(A(s_{\phi(i)})) \cap \phi(I)$ and $\mathrm{Lf}(P(t_i)) \cap I$ do not contain matched leaves, an operation at $A(s_{\phi(i)})$ cannot affect the crossings in $\mathrm{Crs}(P(t_i))$. In the case that there exists $(u, v) \in L(I) \setminus L(I)_0$ with $A(s_{\phi(i)}) \succ (u, v) \succ P(t_i)$, an operation at $A(s_{\phi(i)})$ has already been considered.

Thus, a choice at $P(t_i)$ that extends to a solution must be one that minimizes crossings in $\mathrm{Crs}(P(t_i))$, which is precisely what the algorithm does. The case $P(s_{\phi(i)}) \in M(I)_S$ follows by a similar argument. □

Theorem 49. *`MultiInsertion` solves the Tanglegram Multiple Edge Insertion Problem in $O(2^{|L(I)_1 \cup M(I)_1|}n^2)$ time and $O(n^2)$ space.*

Proof. By Lemma 43, `MultiInsertion` only encounters layouts that restrict to planar layouts of $(T_I, S_{\phi(I)}, \phi|_I)$. By Lemma 48, `MultiInsertion` encounters a solution (X_{min}, Y_{min}) to the Multiple Edge Insertion Problem in some iteration of the **for** loop in line 7. From lines 16-17, we see that the algorithm stores the layout with the fewest number of crossings considered over all iterations of the **for** loop, so it will return either (X_{min}, Y_{min}) or another layout with the same number of crossings. Thus, the output of `MultiInsertion` is a solution to the Multiple Edge Insertion Problem.

For the run-time and space claims, first note that lines 1-6 can all be completed in $O(n^2)$ time. The **for** loop in line 7 then runs for $2^{|L(I)_1 \cup M(I)_1|}$ iterations. The **for** loop in line 9 has at most $2n$ steps, and each step takes $O(n)$ time, as all of the **if** and **else if** conditions can be checked in $O(n)$ time, and paired flips and subtree switches take $O(n)$ time. The remaining steps after line 16 take $O(n^2)$ time for a total of $O(2^{|L(I)_1 \cup M(I)_1|}n^2)$ time. For the space claim, notice that storing all of the sets and layouts takes $O(n^2)$ space. □

Remark 50. In the special case that $|I| = n - 1$, the set $L(I)_1$ is empty and $|M(I)_1| \leqslant 2$. Here, `MultiInsertion` reduces to a less efficient version of `Insertion` that still runs in $O(n^2)$ time.

6 Future work

In Section 3, we defined the flip graph of a planar tanglegram. While paired flips will generate all vertices in this graph, it is possible that some flips do not produce a new layout, as tanglegrams are considered up to isomorphism on T and S. One simple example of this is the unique tanglegram of size 2, where a paired flip at the roots of both trees does not produce a new layout. As such, we pose the following problem.

Problem 1. For any planar tanglegram (T, S, ϕ), characterize the flip graph $\Gamma(T, S, \phi)$. In particular, determine the number of vertices, the number of edges, and the degree of any vertex.

Billey, Konvalinka, and Matsen previously gave an algorithm for generating tanglegrams uniformly at random [2]. We propose a corresponding problem for the case of planar tanglegrams. Since there is a bijection between irreducible planar tanglegrams and pairs of triangulations with no common diagonal from [17], solving this problem may lead to solutions to other open problems.

Problem 2. Construct an efficient algorithm generating planar tanglegrams uniformly at random.

Finally, we pose a problem about using `MultiInsertion` to approximate the tanglegram crossing number, where the bound is modeled after one in [4]. For any tanglegram (T, S, ϕ), this also requires finding a planar subtanglegram $(T_I, S_{\phi(I)}, \phi|_I)$, and from Corollary 40, we know that we do not necessarily want a subtanglegram of maximum size. Additionally, we insist on an efficient algorithm, so we must modify `MultiInsertion` at the vertices in $L(I)_1 \cup M(I)_1$.

Problem 3. Use `MultiInsertion` to construct an efficient algorithm that finds a tanglegram layout with at most $O(\mathrm{crt}(T, S, \phi) \cdot \mathrm{poly}(\log n))$ crossings, where n is the size of (T, S, ϕ) and $\mathrm{poly}(x)$ is some polynomial in x.

Acknowledgements

We would like to thank Sara Billey for proposing tanglegrams as an area of research, suggesting improvements to several proofs, and giving valuable feedback in earlier versions of this paper. We would also like to thank Herman Chau, Paige Helms, Matjaž Konvalinka, Stark Ryan, Stephan Wagner, and the reviewers for valuable feedback in earlier versions of this paper. An extended abstract of this paper previously appeared in [14].

References

[1] R. Anderson, S. Bai, F. Barrera-Cruz, É. Czabarka, G. Da Lozzo, N. L. F. Hobson, J. C.-H. Lin, A. Mohr, H. C. Smith, L. A. Székely, and H. Whitlatch. Analogies between the crossing number and the tangle crossing number. *Electron. J. Combin.*, 25(4):#P4.24, 15, 2018.

[2] S. C. Billey, M. Konvalinka, and F. A. Matsen. On the enumeration of tanglegrams and tangled chains. *J. Combin. Theory Ser. A*, 146:239–263, 2017.

[3] K. Buchin, M. Buchin, J. Byrka, M. Nöllenburg, Y. Okamoto, R. I. Silveira, and A. Wolff. Drawing (complete) binary tanglegrams: hardness, approximation, fixed-parameter tractability. *Algorithmica*, 62(1-2):309–332, 2012.

[4] J. Chuzhoy, S. Mahabadi, and Z. Tan. Towards better approximation of graph crossing number. *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84, 2020.

[5] É. Czabarka, L. A. Székely, and S. Wagner. A tanglegram Kuratowski theorem. *J. Graph Theory*, 90(2):111–122, 2019.

[6] H. Fernau, M. Kaufmann, and M. Poths. Comparing trees via crossing minimization. *J. Comput. Syst. Sci.*, 76:593–608, 2010.

[7] M. R. Garey and D. S. Johnson. Crossing number is np-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.

[8] I. M. Gessel. Counting tanglegrams with species. *J. Comb. Theory, Ser. A*, 184:105498, 2021.

[9] C. Gutwenger, P. Mutzel, and R. Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41:289–308, 2001.

[10] P. Hliňený and G. Salazar. On the crossing number of almost planar graphs. In *Graph Drawing*, pages 162–173, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[11] OEIS Foundation Inc. The on-line encyclopedia of integer sequences. http://oeis.org, 2021.

[12] D. E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., USA, 1997.

[13] M. Konvalinka and S. Wagner. The shape of random tanglegrams. *Adv. Appl. Math.*, 78:76–93, 2016.

[14] K. Liu. Planar tanglegram layouts and single edge insertion. *Séminaire Lotharingien de Combinatoire*, 86B(45), 2022.

[15] A. Lozano, R. Y. Pinter, O. Rokhlenko, G. Valiente, and M. Ziv-Ukelson. Seeded tree alignment and planar tanglegram layout. In *WABI*, pages 98–110, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[16] F. A. Matsen, S. C. Billey, A. Kas, and M. Konvalinka. Tanglegrams: a reduction tool for mathematical phylogenetics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(1):343–349, 2015.

[17] D. Ralaivaosaona, J. B. Ravelomanana, and S. Wagner. Counting Planar Tanglegrams. In *29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2018)*, volume 110 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Appendix

Algorithm 8: `Insertion Case` $v_0 >_S v_{Tmax}$

Input: tanglegram (T, S, ϕ), index i such that $(T_I, S_{\phi(I)}, \phi|_I)$ is planar for $I = [n] \setminus \{i\}$

Output: layout of (T, S, ϕ) that restricts to a planar layout of $(T_I, S_{\phi(I)}, \phi|_I)$

`// Step 1: initialize the algorithm.`

$(X, Y), L :=$ `ModifiedUntangle`$(T, S, \phi|_I)$

construct $L(I)$ from L using Definition 26

$u_0 :=$ parent of t_i, $v_0 :=$ parent of $s_{\phi(i)}$

$L(I)_T := \{(u, v) \in L(I) : u >_T t_i, v \not>_S s_{\phi(i)}\}$

`// Step 2: construct edge sets.`

linearly order $L(I)_T = \{(u_j, v_j)\}_{j=1}^{m}$ so that $u_1 <_T u_2 <_T \ldots <_T u_m$

$E(v_0) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(v_0) \setminus \mathrm{Lf}(v_m)$

$E(u_0) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(u_0) \setminus \{t_i\}$

for $j = 1, 2, \ldots, m,$ **do**

    $E(u_j) :=$ between-tree edges with an endpoint in $\mathrm{Lf}(u_j) \setminus \mathrm{Lf}(u_{j-1})$

`// Step 3: use paired flips and subtree switches to change crossings.`

if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(v_0)$ *in the layout* (X, Y), **then**

    update $Y :=$ `SubtreeSwitch`(Y, v_0)

for $j = m, \ldots, 2, 1,$ **do**

    if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(u_j)$ *in the layout* (X, Y), **then**

        update $(X, Y) :=$ `PairedFlip`$((X, Y), (u_j, v_j))$

if $(t_i, s_{\phi(i)})$ *crosses more than half of the edges in* $E(u_0)$ *in the layout* (X, Y), **then**

    update $X :=$ `SubtreeSwitch`(X, u_0)

return (X, Y)