Non-crossing drawings of multiple geometric Steiner arborescences

Non-crossing drawings of multiple geometric Steiner arborescences*

Irina Kostitsyna† Bettina Speckmann‡ Kevin Verbeek†

1 Introduction

An important problem in the area of computational geometry is the Euclidean Steiner Tree problem: given a set of n points in the plane, find a set of line segments that connect all points in a single connected component, such that the total length of the line segments is minimized. The Euclidean Steiner Tree problem and variants thereof have many applications in practice. For example, rectilinear Steiner trees, where line segments must be horizontal or vertical, are commonly used for wire routing in VLSI design. Also of interest are Steiner arborescences [5]: Steiner trees rooted at a node r, such that the path in the Steiner tree between r and any other input point must be a shortest path with respect to some metric (see Fig. 1). A variant of Steiner arborescences, namely angle-restricted Steiner arborescences, or *flux trees*, have recently been used to design flow maps [4]. In a flux tree, the path from an input point to the root r must always go roughly in the direction of r, that is, it can only deviate by at most a fixed angle.

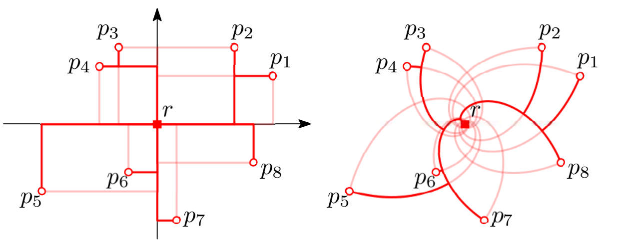

Figure 1: A rectilinear Steiner arborescence and a flux tree with eight terminals.

The Euclidean Steiner tree problem and its variants have been studied extensively. Although most of these problems are NP-hard, many efficient approximation algorithms are known [2, 8]. However, if we want to compute multiple Steiner trees for multiple point sets, such that the Steiner trees have no or few crossings, then there are very few results. Aichholzer *et al.* [1] give an algorithm that, given two sets of n points in the plane, computes in $O(n \log n)$ time two spanning trees (not Steiner trees) such that the diameters of the trees and the number of intersections between the trees are small. Similar (weaker) results have also been obtained for drawing more than two plane spanning trees with few crossings [6, 7]. Recently, Bereg *et al.* [3] presented approximation algorithms for computing k disjoint Steiner trees for k point sets, with approximation ratios $O(\sqrt{n} \log k)$ and $k + \varepsilon$ for general k, $(5/3 + \varepsilon)$ for $k = 3$, and a PTAS for $k = 2$.

In this paper we consider multiple Steiner arborescences. Two or more non-crossing Steiner arborescences need not even exist. Nonetheless, they are very relevant in practice, for example for constructing flow maps. A flux tree can only show information about one source, but ideally multiple sources should be shown simultaneously, in such a way that the corresponding flux trees have few or no crossings. To the best of our knowledge, these problems have not been studied.

Problem statement. We study the following problem: given a set of k roots $r_1, \ldots, r_k \in \mathbb{R}^2$, and k sets of terminals $T_1, \ldots, T_k \subset \mathbb{R}^2$, do there exist k non-crossing Steiner arborescences which connect each set of terminals T_i to its root r_i? We focus mostly on the case $k = 2$. When considering only two trees, we refer to the first tree as the *red* tree, with root r_1 and terminals $T_1 = \{p_1, \ldots, p_n\}$, and the second tree as the *blue* tree with root r_2 and terminals $T_2 = \{q_1, \ldots, q_m\}$. We consider both rectilinear Steiner arborescences and flux trees.

Preliminaries. It follows from the definition of geometric Steiner arborescences that the path between the root and a terminal must completely lie in a particular region. For rectilinear Steiner arborescences this is the rectangle spanned by the root and the terminal. For flux trees this region is bounded by two logarithmic spirals and is hence called the *spiral region* [4]. Here we refer to these regions as $\mathcal{R}$*-regions* and denote the $\mathcal{R}$-region between a root r and a terminal t by $\mathcal{R}(r, t)$. When considering multiple rectilinear Steiner arborescences, we allow the sets of axes of the two Steiner arboresences to be different.

We say that two $\mathcal{R}$-regions $\mathcal{R}(r_1, p_i)$ and $\mathcal{R}(r_2, q_j)$ *fully intersect* if $r_1, p_i \notin \mathcal{R}(r_2, q_j)$, $r_2, q_j \notin \mathcal{R}(r_1, p_i)$, and segments $r_1 p_i$ and $r_2 q_j$ intersect. It is easy to verify that two non-crossing Steiner arborescences do not exist if there are two $\mathcal{R}$-regions $\mathcal{R}(r_1, p_i)$ and $\mathcal{R}(r_2, q_j)$ that fully intersect (see Fig. 2): any two paths routed

*I.K. and B.S. are partially supported by the Netherlands Organisation for Scientific Research (NWO) under grant number 639.023.208. K.V. is supported by the Netherlands Organisation for Scientific Research (NWO) under grant number 639.021.541. I.K. is also supported by F.R.S.-FNRS.

†Computer Science Department, Université libre de Bruxelles, Belgium, irina.kostitsyna@ulb.ac.be

‡Dep. of Mathematics and Computer Science, TU Eindhoven, The Netherlands, [b.speckmann|k.a.b.verbeek]@tue.nl

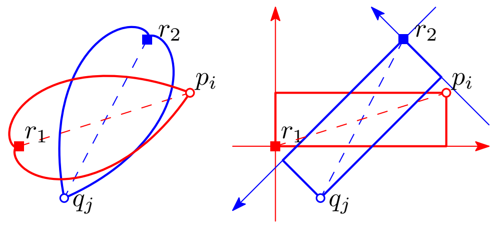

Figure 2: Two $\mathcal{R}$-regions fully intersect.

within the respective $\mathcal{R}$-regions must intersect.

When drawing (the paths of) Steiner arborescences we consider two models:

(a) *Free turns*: Paths can turn anywhere.

(b) *Limited turns*: Paths can only turn at a Steiner point or at a corner of an $\mathcal{R}$-region (as in Fig. 1).

The limited turns model can be quite restrictive. Fig. 3 shows an example where a non-crossing drawing of two rectilinear Steiner arborescences exists only in the free turns model.

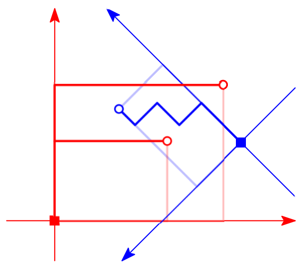

Figure 3: Non-crossing drawing exists only in free turns model.

Results. In the free turns model, we show in Section 2.1 that two rectilinear Steiner arborescences have a non-crossing drawing if (a) no two $\mathcal{R}$-regions fully intersect, and (b) the roots are not contained inside any $\mathcal{R}$-region. In Section 2.2 we lift the constraint on the roots and show how to reduce the decision problem to 2SAT. In Section 3 we show that in the limited turns model it is NP-hard to decide whether multiple rectilinear Steiner arborescences have a non-crossing drawing. The setting of flux trees is more subtle. Our NP-hardness result extends, but testing whether there exists a non-crossing drawing requires additional conditions to be fulfilled (see Section 4). Due to space limitations, some figures and proofs are omitted from this short abstract and can be found in the full version of the paper.

2 Two rectilinear Steiner arborescences

In this section we show how to decide if a non-crossing drawing of two rectilinear Steiner arborescences in the free turns model exists, and how to construct such a drawing. We consider the general case, when the axes of the two arborescences are not aligned. The free turn model implies that, in principle, the paths of the trees can approximate any xy-monotone curve. We show that we can in fact restrict the directions of the paths to the 8 directions implied by the axes of the two rectilinear Steiner arborescences (see Fig. 4).

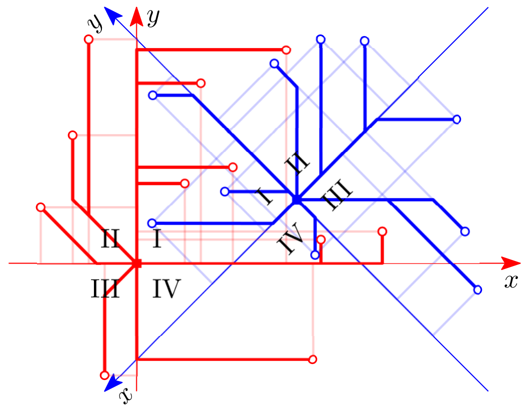

Figure 4: Non-crossing drawing of two rectilinear Steiner arborescences.

2.1 Roots not contained in $\mathcal{R}$-regions

Consider the four quadrants of the coordinate system of the red arborescence ordered counter-clockwise, and the four quadrants of the blue arborescence ordered clockwise. Let the first quadrants face each other (see Fig. 4). There are eleven faces in the arrangement of the four coordinate axes, to which we refer by the two corresponding quadrants. For simplicity of presentation, we assume that no terminal lies on an axis of the other color. Let $\mathcal{C}_b$ be a cone in the red coordinate system with the apex in the blue root and with angle range $[0, \frac{\pi}{2}]$, and let $\mathcal{C}_r$ be a cone in the blue coordinate system with the apex in the red root and with angle range $[0, \frac{\pi}{2}]$. If the roots are not contained in the $\mathcal{R}$-regions of the other tree then there are no red terminals in $\mathcal{C}_b$, and there are no blue terminals in $\mathcal{C}_r$.

Given a red terminal p, and some xy-monotone path π_p connecting p to r_1, define a *dead region* $\mathcal{D}_2(\pi_p)$, with respect to the blue root r_2, to be the union of all points q such that path π_p intersects region $\mathcal{R}(r_2, q)$ and disconnects q from r_2. Analogously, define a *dead region* $\mathcal{D}_1(\pi_q)$ for a blue terminal q.

Observe that π_p is on the boundary of $\mathcal{D}_2(\pi_p)$, and that the rest of its boundary consists of lines parallel to blue axes. For example, in Fig. 5, $\mathcal{D}_2(\pi_p)$ is bounded by two lines parallel to the blue y-axis that go through r_1 and p (as p lies in the blue quadrant II). If p were, for example, in quadrant I, than the bounding line passing through p would be parallel to the blue x-axis.

Given a red terminal p such that $\mathcal{R}(r_1, p)$ does not contain r_2, define the *dead region* $\mathcal{D}_2(p)$ to be the intersection of dead regions $\mathcal{D}_2(\pi_p)$ for all possible paths π_p connecting p to r_1, i.e., $\mathcal{D}_2(p) = \bigcap_{\pi_p} \mathcal{D}_2(\pi_p)$. Define

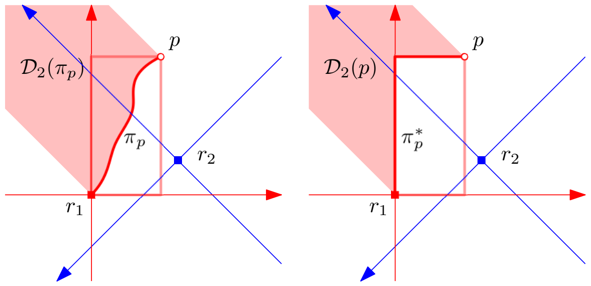

Figure 5: Dead regions of a path π_p and a terminal p.

red terminals in		**blue terminals in**
(a) $\mathrm{I} \cap \mathrm{II}$	vs.	(b) $\mathrm{II} \cap \mathrm{I}$,
(c) $\mathrm{I} \cap \mathrm{III}$	vs.	(d) $\mathrm{III} \cap \mathrm{I}$,
(e) $\mathrm{I} \cap \mathrm{IV}$	vs.	(f) $\mathrm{IV} \cap \mathrm{I}$,
(g) $\mathrm{III} \cap \mathrm{IV}$	vs.	(h) $\mathrm{IV} \cap \mathrm{III}$,
(i) $\mathrm{I} \cap \mathrm{III}$	vs.	(j) $\mathrm{IV} \cap \mathrm{IV}$,
(k) $\mathrm{IV} \cap \mathrm{IV}$	vs.	(l) $\mathrm{III} \cap \mathrm{I}$.

Table 1: Mutually exclusive cases of locations of red and blue terminals.

dead region $\mathcal{D}_1(q)$ analogously. From this definition it follows that:

Proposition 1 *Let a red terminal $p \notin \mathcal{R}(r_2, q)$ and a blue terminal $q \notin \mathcal{R}(r_1, p)$. Then $q \in \mathcal{D}_2(p)$ if and only if $\mathcal{R}(r_1, p)$ fully intersects $\mathcal{R}(r_2, q)$, and therefore $q \in \mathcal{D}_2(p)$ if and only if $p \in \mathcal{D}_1(q)$.*

There can be terminals whose dead regions are empty. For example, if $p \in \mathrm{I} \cap \mathrm{I}$, then there is a path connecting p to r_1 that does not obstruct routing of any possible blue terminal. Consider the eight faces of the axes arrangement except for faces $\mathrm{I} \cap \mathrm{I}$, $\mathrm{I} \cap \mathrm{IV}$, and $\mathrm{IV} \cap \mathrm{I}$. For terminals p and q in them, $\mathcal{D}_2(p)$ and $\mathcal{D}_1(q)$ are not empty. Moreover, in these faces $p \in \mathcal{D}_2(p)$ and $q \in \mathcal{D}_1(q)$. Denote π_p^* to be the path that connects p to r_1 along the boundary of $\mathcal{D}_2(p)$, and π_q^* to be the path that connects q to r_2 along the boundary of $\mathcal{D}_1(q)$ (see Fig. 5 (right)). We can show that:

Proposition 2 *Paths π_p^* and π_q^* are xy-monotone in the red and blue coordinate systems, respectively.*

Therefore π_p^* and π_q^* are valid paths connecting p to r_1 and q to r_2. From Proposition 1 it follows that if a blue terminal $q \notin \mathcal{D}_2(p)$ then π_p^* does not itersect π_q^*. In the full version of the paper we carefully go through all cases for terminals p and q such that the corresponding dead regions $\mathcal{D}_2(p)$ and $\mathcal{D}_1(q)$ are not empty and their boundaries contain paths connecting the terminals to their roots.

Routing rules. Notice that two cases, when there is a red terminal p in $\mathrm{I} \cap \mathrm{II}$, and when there is a blue terminal q in $\mathrm{II} \cap \mathrm{I}$, are mutually exclusive, for otherwise $\mathcal{R}(r_1, p)$ would fully intersect $\mathcal{R}(r_2, q)$. Table 1 gives a full list of all mutually exclusive cases. Given two roots and two sets of terminals such that no two $\mathcal{R}$-regions of opposite colors fully intersect, we can construct two non-intersecting Steiner arborescences using simple routing rules (see Fig. 4). First, red terminals p in $(\mathrm{II} \cup \mathrm{III}) \setminus \mathcal{C}_r$, $\mathrm{I} \cap \mathrm{II}$, $\mathrm{I} \cap \mathrm{III}$, $\mathrm{IV} \cap \mathrm{III}$, or $\mathrm{IV} \cap \mathrm{IV}$ are routed along π_p^*. Blue terminals q in $(\mathrm{II} \cup \mathrm{III}) \setminus \mathcal{C}_b$, $\mathrm{II} \cap \mathrm{I}$, $\mathrm{III} \cap \mathrm{I}$, $\mathrm{III} \cap \mathrm{IV}$, or $\mathrm{IV} \cap \mathrm{IV}$ are routed along π_q^*. Next, terminals in $\mathcal{C}_r$ and $\mathcal{C}_b$ are routed as shown in Fig. 6. Lastly, the rest of the terminals are routed so as to avoid already constructed paths. Detailed routing rules can be found in the full version.

Theorem 3 *Two rectilinear Steiner arborescences can be drawn with no crossings in the free turn model if no two $\mathcal{R}$-regions fully intersect and if no roots are contained in $\mathcal{R}$-regions.*

2.2 Roots contained in $\mathcal{R}$-regions

Next, we relax the restriction that the roots cannot be contained in $\mathcal{R}$-regions. Now, for any $\mathcal{R}$-region that contains the root of the other color, we need to make a choice of how to route the terminal-to-root path around the other root. This choice clearly can affect later decisions. Before we proceed, we need some additional definitions.

Points r and t split the boundary of $\mathcal{R}(r, t)$ into two components that we call the *right* side $\sigma^+(r, t)$ (that leaves the $\mathcal{R}$-region to the left if moving from r to t), and the *left* side $\sigma^-(r, t)$.

We say that $\mathcal{R}(r_1, p_i)$ *cuts* the right (left) side of $\mathcal{R}(r_2, q_j)$, if $r_1 \in \mathcal{R}(r_2, q_j)$, and both sides of $\mathcal{R}(r_1, p_i)$ intersect the right (left) side of $\mathcal{R}(r_2, q_j)$ (see Fig. 7). We can now define a dead region for a terminal p for a fixed direction a p-to-r_1 path must take around r_2.

Given a red terminal p, define the *left (right) dead region* $\mathcal{D}_2^l(p)$ ($\mathcal{D}_2^r(p)$) to be the intersection of dead regions $\mathcal{D}_2(\pi_p)$ for all possible paths π_p connecting p to r_1 that go around r_2 from the left (right), i.e., $\mathcal{D}_2^l(p) = \bigcap_{\text{left } \pi_p} \mathcal{D}_2(\pi_p)$ and $\mathcal{D}_2^r(p) = \bigcap_{\text{right } \pi_p} \mathcal{D}_2(\pi_p)$. Analogously, define $\mathcal{D}_1^l(q)$ and $\mathcal{D}_1^r(q)$. Note that in this definition we do not require $\mathcal{R}$-regions to be root free. We can make an observation similar to the one in the

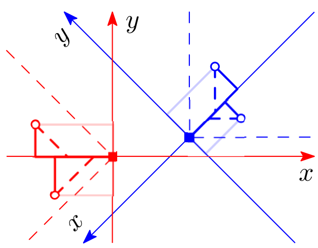

Figure 6: Routing rule for terminals in $\mathcal{C}_r$ and $\mathcal{C}_b$.

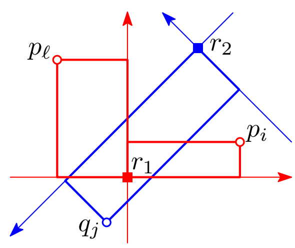

Figure 7: Red $\mathcal{R}$-regions cut the left and right side of the blue $\mathcal{R}$-region.

previous section. Let the blue root $r_2 \in \mathcal{R}(r_1, p)$ and the red root $r_1 \notin \mathcal{R}(r_2, q)$. A blue terminal $q \in \mathcal{D}_2^l(p)$ ($q \in \mathcal{D}_2^r(p)$) if and only if $\mathcal{R}(r_2, q)$ fully intersects the left side $\sigma^+(r_1, p)$ (right side $\sigma^-(r_1, p)$) of $\mathcal{R}(r_1, p)$. Therefore, if $r_2 \in \mathcal{R}(r_1, p)$ and $r_1 \notin \mathcal{R}(r_2, q)$, the blue terminal $q \notin \mathcal{D}_2^l(p)$ ($q \notin \mathcal{D}_2^r(p)$) if and only if $\mathcal{D}_1(q)$ does not intersect $\mathcal{D}_2^l(p)$ ($\mathcal{D}_2^r(p)$). We can extend this observation to the case where the blue root $r_2 \in \mathcal{R}(r_1, p)$ and the red root $r_1 \in \mathcal{R}(r_2, q)$:

Observation 1 *If $r_2 \in \mathcal{R}(r_1, p)$ and $r_1 \in \mathcal{R}(r_2, q)$, the blue terminal $q \notin \mathcal{D}_2^l(p)$ ($q \notin \mathcal{D}_2^r(p)$) if and only if $\mathcal{D}_1^l(q)$ ($\mathcal{D}_1^r(q)$) does not intersect $\mathcal{D}_2^l(p)$ ($\mathcal{D}_2^r(p)$).*

We reduce the problem of choosing the direction of the path with respect to the other root to 2SAT. Given a solution to the 2SAT formula that fixes directions of the paths with terminals in cones $\mathcal{C}_r$ and $\mathcal{C}_b$, we can again route the paths along the boundaries of the dead regions. More details can be found in the full version of the paper.

Theorem 4 *We can decide in polynomial time whether two rectilinear Steiner arborescences can be drawn with no crossings in the free turn model.*

3 Drawing many Steiner arborescences is NP-hard

If the number of arborescences in the problem is not bounded, the problem becomes NP-hard for the limited turns model.

Theorem 5 *It is NP-hard to decide whether k rectilinear Steiner arborescences, where k is part of the input, can be drawn without crossings in the limited turns model, even if all trees are axis-aligned.*

Theorem 6 *It is NP-hard to decide whether k flux trees, where k is part of the input, can be drawn without crossings in the limited turns model.*

4 Two flux trees

In this section we sketch how to draw two flux trees in the free turns model with no root containment in $\mathcal{R}$-regions. Similarly to the rectilinear case, free turns imply that the terminal-to-root paths can approximate any *spiral monotone*[1] curve. Here we restrict the paths to only follow four logarithmic spirals, positive and negative spirals with the origin in the red root, and positive and negative logarithmic spirals with the origin in the blue root. We prove the following theorem in the full version of the paper.

[1] A spiral monotone curve [4] requires that for any point the angle between the tangent and the direction to the destination is not greater than a given parameter α.

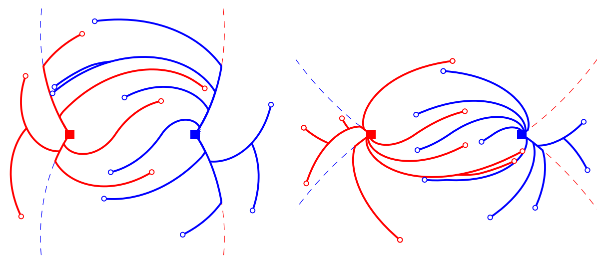

Figure 8: Two non-crossing drawings of flux trees for $\alpha = 60°$ (left) and $\alpha = 30°$ (right).

Theorem 7 *We can decide in polynomial time if two flux trees with no root containment in $\mathcal{R}$-regions can be drawn without crossings in the free turns model.*

Figure 8 shows the final result of the procedure.

References

[1] O. Aichholzer et al. Connecting colored point sets. *Discrete Applied Mathematics*, 155(3), 2007.

[2] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5), 1998.

[3] S. Bereg et al. Colored non-crossing Euclidean Steiner forest. In *ISAAC*, 2015.

[4] K. Buchin, B. Speckmann, and K. Verbeek. Angle-restricted Steiner arborescences for flow map layout. *Algorithmica*, 72(2), 2015.

[5] J. Cong, A. Kahng, and Kwok-Shing Leung. Efficient algorithms for the minimum shortest path Steiner arborescence problem with applications to VLSI physical design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(1), 1998.

[6] M. Kano, C. Merino, and J. Urrutia. On plane spanning trees and cycles of multicolored point sets with few intersections. *Information Processing Letters*, 93(6), 2005.

[7] J. Leaños et al. Spanning trees of multicoloured point sets with few intersections. In *IJCCGGT*, 2005.

[8] B. Lu and L. Ruan. Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. *Journal of Combinatorial Optimization*, 4(3), 2000.