Non-Crossing Geometric Steiner Arborescences

Irina Kostitsyna*[1], Bettina Speckmann†[2], and Kevin Verbeek‡[3]

1 **TU Eindhoven, Eindhoven, The Netherlands**
`i.kostitsyna@tue.nl`
2 **TU Eindhoven, Eindhoven, The Netherlands**
`b.speckmann@tue.nl`
3 **TU Eindhoven, Eindhoven, The Netherlands**
`k.a.b.verbeek@tue.nl`

Abstract

Motivated by the question of simultaneous embedding of several flow maps, we consider the problem of drawing multiple geometric Steiner arborescences with no crossings in the rectilinear and in the angle-restricted setting. When terminal-to-root paths are allowed to turn freely, we show that two rectilinear Steiner arborescences have a non-crossing drawing if neither tree necessarily completely disconnects the other tree and if the roots of both trees are "free". If the roots are not free, then we can reduce the decision problem to 2SAT. If terminal-to-root paths are allowed to turn only at Steiner points, then it is NP-hard to decide whether multiple rectilinear Steiner arborescences have a non-crossing drawing. The setting of angle-restricted Steiner arborescences is more subtle than the rectilinear case. Our NP-hardness result extends, but testing whether there exists a non-crossing drawing if the roots of both trees are free requires additional conditions to be fulfilled.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Steiner arborescences, non-crossing drawing, rectilinear, angle-restricted

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.54

1 Introduction

Flow maps are used in cartography to visualize the movement of objects between different locations. Generally, multiple sources are connected to multiple destinations with curves of varying thickness indicating the amount of flow. A good layout of a flow map consists of "simple" aesthetically pleasing curves, and avoids unnecessary intersections. Specifically, in this paper we are interested in drawing flow maps with no crossings at all. That is, given a set of source points and a set of destination points, we are looking to connect the source points to their corresponding destinations without intersections. For the sake of readability, the curves of a flow map should roughly be oriented from the source to the destination (or vice versa). This poses restrictions on the curves, which in related work [5] has been formalized using the notion of *angle-restricted* paths: for every point p on the path, the angle γ between the tangent vector at p and the vector from p to the source can be at most a prescribed angle α (see Figure 1).

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

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 54; pp. 54:1–54:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

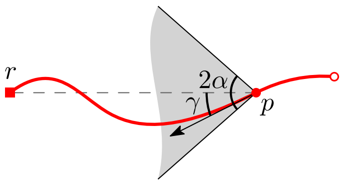


Figure 1 Angle-restricted path: γ is bounded by α.

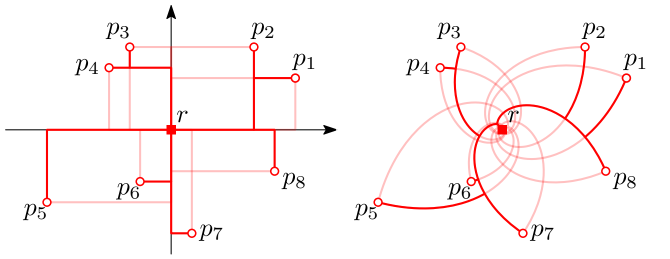


Figure 2 A rectilinear Steiner arborescence and a flux tree with eight terminals.

The notion of angle-restricted paths is closely related to that of so-called *generalized self-approaching* paths. A φ-self-approaching path, as defined by Aichholzer et al. in [2], is a path such that for any point p on it, the rest of the path lies inside some wedge with apex in p and with angle φ. Accordingly, any angle-restricted path with angular constraint α is also generalized self-approaching for angle $\varphi = 2\alpha$: for every point p on the path from some destination to its source r, the remainder of the path is contained in a wedge with angle 2α, with apex at p, and with r lying on its bisector. On the other hand, any α-self-approaching path is also angle-restricted with constraint α.

In a way, angle-restricted paths behave similarly to (xy-monotone) rectilinear paths. In this case, for every point p on the path, the subpath between p and the source r is bounded by an axis-aligned $90°$ wedge with the apex at p that contains r. We therefore study the following problems in this paper. Given a set of source points and a set of corresponding destination points, is it possible to connect all sources to their respective destinations using only angle-restricted paths (or xy-monotone paths in the rectilinear case) such that there are no intersections? The rectilinear case offers a somewhat simpler setting that is more amenable to analysis and allows us to clearly illustrate our main techniques.

In practice, a flow map will often have a small number of source points connected to multiple destination points to show the comparison of a certain commodity flow between several geographic locations, or to compare several types of commodities. Thus, we can group the flows in a flow map by a common source point to represent the out-flow of a given commodity from a specific location. Our problem is equivalent to drawing non-intersecting flow trees with angle-restricted (or xy-monotone for rectilinear) leaf-to-root paths. This approach has an important advantage of considering relevant flows together, and thus allowing for the possibility of merging similar flows which are going to the same source. A resulting merge point will then be a Steiner point, and the flow tree will be a Steiner *arborescence* (a tree with directed edges in the direction from the root). Buchin et al. [5] introduced angle-restricted Steiner arborescences, or *flux trees*, as a new variant of drawing flow trees. They study the problem of drawing a flux tree of minimal total length, and, among other results, show that the branches of an optimal flux tree consist of arcs of logarithmic spirals. Figure 2 shows an example of a rectilinear Steiner arborescence and a flux tree.

For two or more sets of input points, non-crossing Steiner arborescences need not even exist. Nonetheless, they are very relevant in practice. A single flux tree can show information about only 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 yet been studied. In this paper we are therefore studying the decision question of whether there exists a simultaneous non-crossing drawing of multiple geometric Steiner arborescences. Specifically, given a set of k roots (sources) $r_1, \dots, r_k \in \mathbb{R}^2$, and k sets of terminals (destinations) $T_1, \ldots, T_k \subset \mathbb{R}^2$, do there exist k non-crossing Steiner arborescences which connect each set of terminals T_i to its root r_i, such that the leaf-to-root paths are angle-restricted (or xy-monotone for rectilinear)?

When talking about flows in a flow map, we assume the (standard in the literature) root-to-terminal orientation of the flows[1]. However, when drawing a flow tree, we start from a terminal and move towards the root. Thus, the direction of the paths in the construction of the trees is opposite to the flows in the flow map. When needed, we will explicitly state which orientation is considered to avoid any ambiguity.

Related work. 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 [3, 10]. 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 [7, 9]. Recently, Bereg *et al.* [4] 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 $k = 2$. Other relevant related work considers obtaining particular subgraphs of given geometric or topological graphs with few or no crossings [6, 8, 11, 12]. These problems are often NP-hard, except for certain special cases [12], but they differ from general Steiner tree problems, as the selected edges must be part of the input graph.

Preliminaries. In this paper, we focus mostly on the case of drawing two flow trees, i.e., when $k = 2$. 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 $T_2 = \{q_1, \ldots, q_m\}$. When studying multiple rectilinear Steiner arborescences, we generally allow the use of different axes for different trees. This way, the rectilinear problem is more similar to the more involved angle-restricted (flux trees) version of the problem.

It follows from the restriction on the paths that the path between a terminal and its root must completely lie in a particular region. For rectilinear Steiner arborescences this is the axes-aligned rectangle spanned by the root and the terminal. For flux tree this region is bounded by two curves traced by points for which the angle between the tangent and the direction to the destination is exactly α. These curves are in fact logarithmic spirals, and hence the above region is called the *spiral region* [5] (see Figure 3). Here we refer to these regions as $\mathcal{R}$-*regions*, and denote the $\mathcal{R}$-region given by a root r and a terminal t by $\mathcal{R}(r, t)$.

▶ **Definition 1.** 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 for $p_i \in T_1$ and $q_j \in T_2$ (see Figure 3): any two paths routed within the respective $\mathcal{R}$-regions must intersect.

[1] The same flow map can as well be used to represent an in-flow of a product with the flows oriented from terminals to roots.

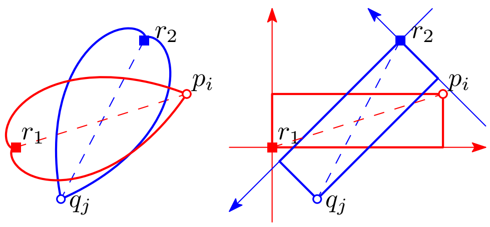


Figure 3 Two $\mathcal{R}$-regions fully intersect.

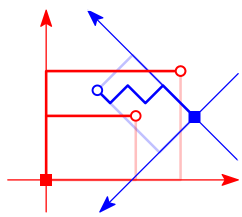


Figure 4 Limited turns: no drawing.

When drawing (the paths of) Steiner arborescences, we will consider two models. In the *limited turns* model, we restrict the segments of a terminal-to-root path of a flux tree to only follow one of the two logarithmic spirals implied by the location of the corresponding root, and prohibit the path from making a turn anywhere except for maybe at a merging point with another path. Similarly, in the *limited turns* model in the rectilinear case, we restrict terminal-to-root paths to be rectilinear, and prohibit the paths from making a turn except for at a merging point with another path (or at a corner of an $\mathcal{R}$-region). In the *free turns* model, we allow the paths to follow any angle-restricted (or xy-monotone) curves as long as there is no crossings. The limited turns model can be quite restrictive. Figure 4 shows an example where a non-crossing drawing of two rectilinear Steiner arborescences exists only if free turns are allowed.

Results. In Section 2.1 we show that two rectilinear Steiner arborescences, in the case when the roots are not contained inside any $\mathcal{R}$-regions, have a non-crossing drawing in the free turns model if and only if no two $\mathcal{R}$-regions fully intersect. In Section 2.2 we lift the constraint on the roots and show how to reduce the decision problem to 2SAT. For flux trees the problem is more involved: $\mathcal{R}$-regions of flux trees can have more complicated interactions. Contrary to rectilinear Steiner arborescences, it is not sufficient for flux trees to consider only full intersections of $\mathcal{R}$-regions if the roots are not contained in $\mathcal{R}$-regions. Nonetheless we can extend our arguments for rectilinear Steiner arborescences to show that, in the case when the roots are not contained in $\mathcal{R}$-regions, we can decide in polynomial time if two flux trees have a non-crossing drawing. Due to space constraints we provide only a sketch of this extension in Section 3; refer to the full version of this paper for details. In the limited turns model we can show that it is NP-hard to decide whether an arbitrary number of rectilinear Steiner arborescences or flux trees have a non-crossing drawing. This result, as well as all the omitted proofs, can be found in the full version of this 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 if the answer is positive. We consider the general case, when the axes of the 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 restrict the directions of the paths to the 8 directions implied by the axes of the two rectilinear Steiner arborescences.

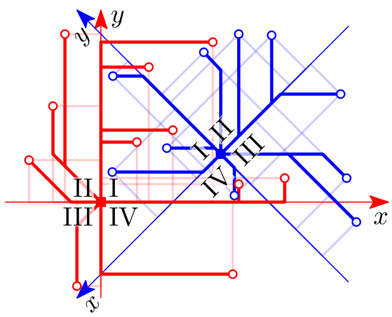


Figure 5 Non-crossing drawings of two Steiner arborescences.

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

Consider the four quadrants of the coordinate system of the red arborescence ordered counterclockwise, and the four quadrants of the blue arborescence ordered clockwise. Let the first quadrants be the ones containing the other root (see Figure 5). In the arrangement of the four coordinate axes there are eleven faces, 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 with angle range $[0, \frac{\pi}{2}]$ in the red coordinate system with the apex in the blue root, and let $\mathcal{C}_r$ be a cone with angle range $[0, \frac{\pi}{2}]$ in the blue coordinate system with the apex in the red root. 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 $\mathcal{R}(q, r_2)$ and disconnects q from r_2. Analogously, given a blue terminal q and some xy-monotone path π_q connecting q to r_2, define a *dead region* $\mathcal{D}_1(\pi_q)$ to be the union of all points p such that path π_q disconnects p from r_1 in region $\mathcal{R}(p, r_1)$.

Observe that π_p is on the boundary of $\mathcal{D}_2(\pi_p)$, and that the rest of the boundary consists of lines parallel to blue axes. For example, in Figure 6, $\mathcal{D}_2(\pi_p)$ is bounded on one side by a line that goes through r_1 that is parallel to the blue y-axis. On the other side $\mathcal{D}_2(\pi_p)$ is bounded by a line parallel to the blue y-axis that goes through p, as p is in the blue quadrant II. If p were, for example, in blue quadrant I, than the bounding line would be parallel to blue x-axis. Also note that there are terminals p such that $\mathcal{D}_2(\pi_p)$ only consists of the points of π_p.

▶ **Definition 2.** Given a red terminal p such that $r_2 \notin \mathcal{R}(p, r_1)$, define the *dead region* $\mathcal{D}_2(p)$ with respect to r_2 to be the intersection of dead regions $\mathcal{D}_2(\pi_p)$ for all possible paths π_p connecting p to r_1:

$$\mathcal{D}_2(p) = \bigcap_{\pi_p} \mathcal{D}_2(\pi_p)\,.$$

Define the *dead region* $\mathcal{D}_1(q)$ of a blue terminal q analogously.

▶ **Proposition 3.** *For a red terminal $p \notin \mathcal{R}(q, r_2)$ and a blue terminal $q \notin \mathcal{R}(p, r_1)$, the following three statements are equivalent: (a) $q \in \mathcal{D}_2(p)$, (b) $p \in \mathcal{D}_1(q)$, (c) $\mathcal{R}(p, r_1)$ and $\mathcal{R}(q, r_2)$ fully intersect.*

Note that there can exist 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

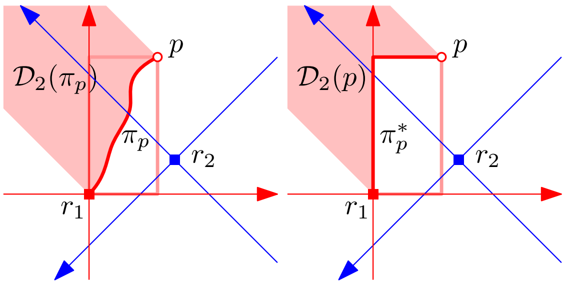
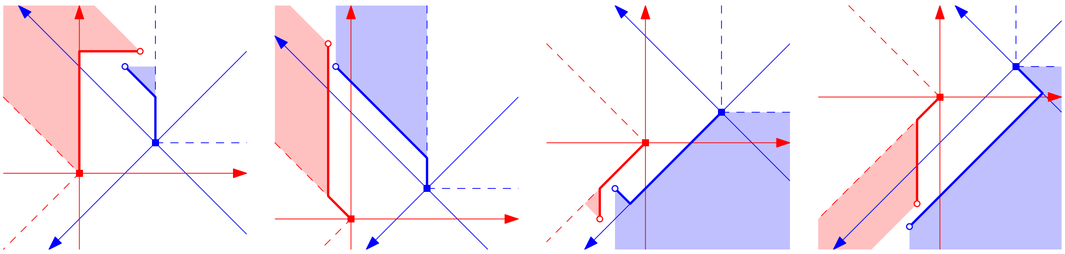


Figure 6 Dead regions: of a path π_p (left) and of a terminal p (right).



(a) $\mathrm{I} \cap \mathrm{II}$. **(b)** $\mathrm{II} \cap \mathrm{II}$. **(c)** $\mathrm{III} \cap \mathrm{I}$. **(d)** $\mathrm{III} \cap \mathrm{IV}$.

Figure 7 Red terminal p, blue terminal q, the corresponding dead regions $\mathcal{D}_2(p)$ (light red) and $\mathcal{D}_1(q)$ (light blue), and paths π_p^* and π_q^* connecting p to r_1 and q to r_2. Cones $\mathcal{C}_r$ and $\mathcal{C}_b$ are denoted with dashed red and blue lines respectively.

terminal. Consider the eight faces of the arrangement of the four axes 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 $\mathcal{D}_2(p)$ (refer to Figure 6 (right)). Similarly, denote π_q^* to be the path that connects q to r_2 along the boundary of $\mathcal{D}_1(q)$. We can show that:

▶ **Proposition 4.** *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 3 it follows that if a blue terminal $q \notin \mathcal{D}_2(p)$ then π_p^* does not intersect π_q^*. Figure 7 illustrates some of the possible placements of p and q such that their dead regions $\mathcal{D}_2(p)$ and $\mathcal{D}_1(q)$ are not empty.

Routing rules. Note 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. Otherwise there is no crossing free drawing of the arborescences. Table 1 gives a full list of all mutually exclusive cases. We will prove that, given two roots and two sets of terminals such that no two $\mathcal{R}$-regions of opposite colors fully intersect, there exists a non-crossing drawing of two Steiner arborescences. We can draw two non-crossing Steiner arborescences using the following routing rules:

Rule 1. If a red terminal $p \in (\mathrm{II} \cup \mathrm{III}) \backslash \mathcal{C}_r$ (refer to Figure 7 (b, c, d)), or p is in faces $\mathrm{I} \cap \mathrm{II}$, $\mathrm{I} \cap \mathrm{III}$, $\mathrm{IV} \cap \mathrm{III}$, or $\mathrm{IV} \cap \mathrm{IV}$ (refer to Figure 7 (a)), then connect p to r_1 along π_p^*.

Rule 2. If a blue terminal $q \in (\mathrm{II} \cup \mathrm{III}) \backslash \mathcal{C}_b$ (refer to Figure 7 (a, b)), or q is in faces $\mathrm{II} \cap \mathrm{I}$, $\mathrm{III} \cap \mathrm{I}$, $\mathrm{III} \cap \mathrm{IV}$, or $\mathrm{IV} \cap \mathrm{IV}$ (refer to Figure 7 (c, d) respectively), then connect q to r_2 along π_q^*.

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

red terminals		blue terminals
(a) in I ∩ II	vs.	(b) in II ∩ I,
(c) in I ∩ III	vs.	(d) in III ∩ I,
(e) in I ∩ IV	vs.	(f) in IV ∩ I,
(g) in III ∩ IV	vs.	(h) in IV ∩ III,
(i) in I ∩ III	vs.	(j) in IV ∩ IV,
(k) in IV ∩ IV	vs.	(l) in III ∩ I.

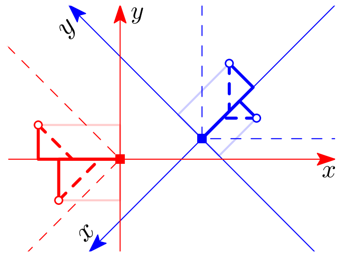
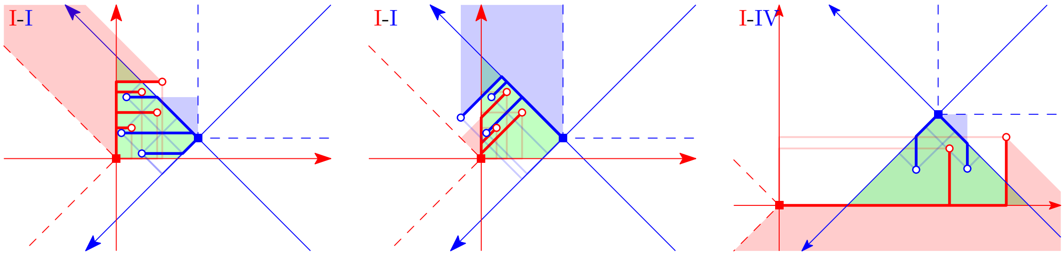


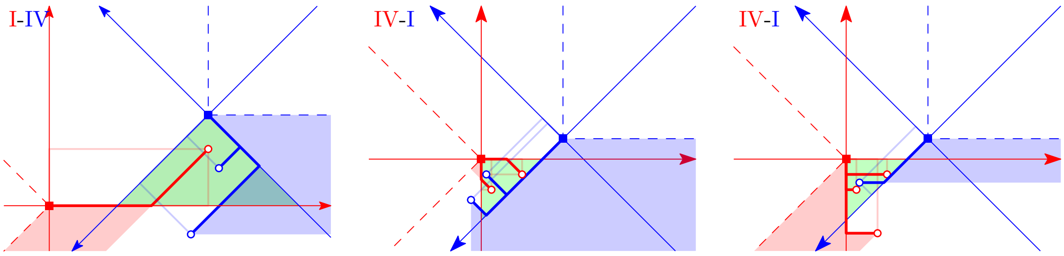
Figure 8 Rule 3.



(a) routing in I ∩ I when ∃ red terminals in I ∩ II

(b) routing in I ∩ I when ∃ blue terminals in II ∩ I

(c) routing in I ∩ IV when ∃ red terminals in I ∩ III



(d) routing in I ∩ IV when ∃ blue terminals in IV ∩ IV

(e) routing in IV ∩ I when ∃ blue terminals in III ∩ I

(f) routing in IV ∩ I when ∃ red terminals in IV ∩ IV

Figure 9 Routing rules for drawing rectilinear Steiner arborescences. Shaded regions denote dead regions.

Rule 3. Route the red terminals in cone $\mathcal{C}_r$ parallel to the red y-axis until reaching the x-axis, then along it. Route the blue terminals in $\mathcal{C}_b$ parallel to the blue y-axis until the x-axis, then along it. For aesthetics, we can add a shortcut in the direction of one of the axes of the opposite color (see Figure 8).

After applying Rules 1–3, all the terminals outside of I ∩ I, I ∩ IV, and IV ∩ I are connected to the roots. We use the following routing rules for the remaining terminals (see Figure 9).

Rule 4. Face I ∩ I: in case (a) (in Table 1), red edges are drawn parallel to the red x-axis until the red y-axis, then follow it, blue edges are drawn parallel to the red x-axis, until a blue axis, then follow it to the root; in case (b), blue edges are drawn parallel to the blue x-axis until the blue y-axis, then follow it, red edges are drawn parallel to the blue x-axis until a red axis, then follow it.

Rule 5. Face I ∩ IV: in case (i) (in Table 1), red edges are drawn parallel to the red y-axis then along the red x-axis, blue edges are drawn parallel to the red y-axis then along the blue y-axis; in case (j), red edges are drawn parallel to the blue x-axis then along the red x-axis, blue edges are drawn parallel to the blue x-axis then along the blue y-axis.

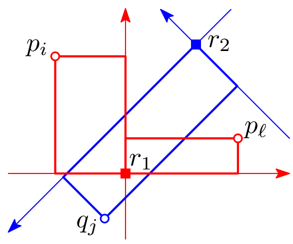


Figure 10 Definition 6.

Rule 6. Face IV ∩ I: in case (k) (in Table 1), red edges are drawn parallel to the red x-axis then along the red y-axis, blue edges are drawn parallel to the red x-axis then along the blue x-axis; in case (l), blue edges are drawn parallel to the blue y-axis then along the blue x-axis, red edges are drawn parallel to the blue y-axis then along a red axis.

Theorem 5 now follows from the definition of the dead regions and a case analysis over faces containing red and blue terminals.

▶ **Theorem 5.** *If the roots are not contained in $\mathcal{R}$-regions, then two rectilinear Steiner arborescences can be drawn with no crossings in the free turn model if and only if no two $\mathcal{R}$-regions fully intersect.*

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

We now relax the restriction that the roots cannot be contained in $\mathcal{R}$-regions. Hence, 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 introduce some additional definitions. Points r and t split the boundary of $\mathcal{R}(t, r)$ into two pieces that we call the *left* and the *right* sides (with respect to moving from t to r).

▶ **Definition 6.** We say that $\mathcal{R}(p_i, r_1)$ cuts the left (right) side of $\mathcal{R}(q_j, r_2)$, if $r_1 \in \mathcal{R}(q_j, r_2)$, and both sides of $\mathcal{R}(p_i, r_1)$ intersect the left (right) side of $\mathcal{R}(q_j, r_2)$ (refer to Figure 10).

We can define a dead region of a terminal p for a fixed direction a p-to-r_1 path must take around r_2:

▶ **Definition 7.** A left (right) dead region $\mathcal{D}_2(p, \text{left})$ ($\mathcal{D}_2(p, \text{right})$) with respect to r_2, for a given red terminal p such that $r_2 \in \mathcal{R}(p, r_1)$, is the intersection of dead regions $\mathcal{D}_2(\pi_p)$ for all possible paths π_p connecting r_1 to p that pass between r_2 and the left (right) side of $\mathcal{R}(p, r_1)$:

$$\mathcal{D}_2(p, \text{left}) = \bigcap_{\text{left } \pi_p} \mathcal{D}_2(\pi_p)\,, \qquad \mathcal{D}_2(p, \text{right}) = \bigcap_{\text{right } \pi_p} \mathcal{D}_2(\pi_p)\,.$$

Analogously, define $\mathcal{D}_1(q, \text{left})$ and $\mathcal{D}_1(q, \text{right})$. Note that we can make a similar observation for left and right dead regions as for dead regions. Let blue root $r_2 \in \mathcal{R}(p, r_1)$. A blue terminal q lies in $\mathcal{D}_2(p, \text{left})$ ($\mathcal{D}_2(p, \text{right})$) if and only if $\mathcal{R}(q, r_2)$ cuts the left (right) side of $\mathcal{R}(p, r_1)$.

We reduce the problem of choosing the direction of the path with respect to the other root by reducing it to 2SAT. We assign a boolean variable to each $\mathcal{R}$-region containing the root of the other color, which takes its value according to the direction in which the terminal-to-root

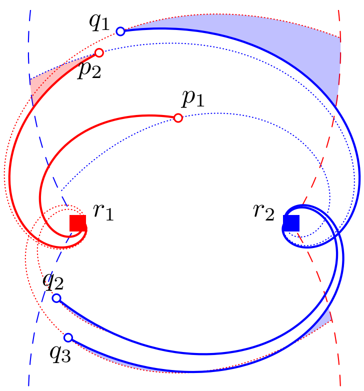


Figure 11 Dead regions for two flux trees.

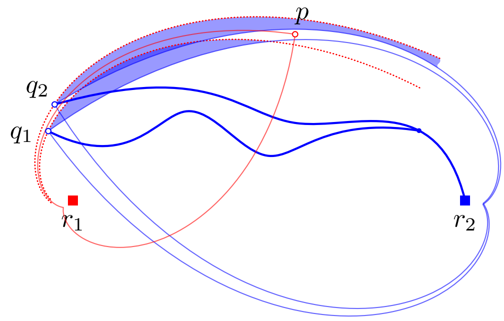


Figure 12 $\mathcal{R}(p, r_q)$ does not fully intersect $\mathcal{R}(q_1, r_2)$ nor $\mathcal{R}(q_2, r_2)$. Nevertheless, there is no angle-restricted path from p to r_1 that does not intersect paths from q_1 and q_2 to r_2. Areas highlighted in light-blue are the dead regions of q_1 and q_2.

path goes around the other root. Given a 2SAT formula solution, we can apply the routing rules and connect the terminals to their roots along the boundaries of the dead regions.

▶ **Theorem 8.** *We can decide in polynomial time whether two rectilinear Steiner arborescences can be drawn without crossings in the free turn model.*

3 Two flux trees

In this section we sketch how to draw two flux trees with no root containment in $\mathcal{R}$-regions in the model when free turns are allowed. The details can be found in the full version of this paper. Similarly to the rectilinear case, free turns imply that a terminal-to-root path can be any angle-restricted curve. Any angle-restricted curve can be approximated with a curve following only four types of logarithmic spirals: left-handed and right-handed, or simply left and right, spirals (left spirals spiral in clockwise direction when moving towards the root, right spirals spiral in counter-clockwise direction) with their origins in the red and blue roots. Thus we can restrict our drawing to these four types of spirals.

Similarly to the rectilinear case, we define the areas $\mathcal{C}_r$ and $\mathcal{C}_b$ which should be empty of blue and red terminals respectively (to fulfill the no-root-containment requirement). These areas are bounded by the spirals centered at one root and going through the other root.

Analogously to the rectilinear case, we can define a dead region of a path, and a dead region of a terminal point. Figure 11 shows an example of several terminals and their dead regions. The dead regions are bounded by two logarithmic spirals going through a terminal and centered at the two roots. Consider, for example, red terminal p_2 in Figure 11. Part of the blue spiral that goes through p_2 is hidden from root r_2 by the red spiral connecting p_2 to r_1. Therefore, for any terminal q above the red spiral, but below the blue spiral (area shaded light-red in the figure), $\mathcal{R}(q, r_2)$ will fully intersect $\mathcal{R}(p_2, r_1)$.

A red and a blue logarithmic spiral can intersect more than once inside the area $\mathbb{R}^2 \setminus (\mathcal{C}_r \cup \mathcal{C}_b)$. This fact can cause some dead regions to consist of several connected components (for example, blue terminal q_3 in Figure 11). Moreover, we no longer can consider the dead regions independently, as we did in the rectilinear setting. Consider the example in Figure 12. Point p does not belong to the dead region of q_1 nor of q_2 ($\mathcal{R}(p, r_1)$ does not fully intersect $\mathcal{R}(q_1, r_2)$ nor $\mathcal{R}(q_2, r_2)$). Nevertheless, no angle-restricted path connecting p to r_1 can avoid paths from q_1 and q_2 to r_2. Indeed, any angle-restricted path from p to r_1 will intersect

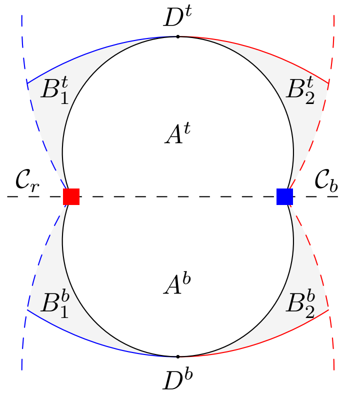


Figure 13 Regions A^t, A^b, B_1^t, B_2^t, B_1^b, B_2^b, D^t, and D^b.

either the dead region of q_1 or the dead region of q_2. Thus, when two or more dead regions intersect, they block some area outside of them that becomes forbidden for the terminals of the other color. We will call this area an *extended dead region*.

To formally define the extended dead region, we need to introduce some notation. Let $s_r^+(p)$ and $s_r^-(p)$ be respectively the spiral segments of the right and left logarithmic spirals, centered at r_1 and going through p, which are bounded by $\mathbb{R}^2 \backslash (\mathcal{C}_r \cup \mathcal{C}_b)$; and let $s_b^+(q)$ and $s_b^-(q)$ be respectively the spiral segments of the right and left logarithmic spirals, centered at r_2 and going through q, which are bounded by $\mathbb{R}^2 \backslash (\mathcal{C}_r \cup \mathcal{C}_b)$. Let $\mathbb{R}^2_+$ be the half-plane to the left of r_1r_2, and $\mathbb{R}^2_-$ be the half-plane to the right of r_1r_2.

▶ **Definition 9.** Given k blue terminals $Q = \{q_1, q_2, \ldots, q_k\}$ in $\mathbb{R}^2_+ \backslash (\mathcal{C}_r \cup \mathcal{C}_b)$, such that the component of the dead region $\mathcal{D}_1(q_i)$ containing q_i intersects the left side of $\mathcal{R}(q_{i+1}, r_2)$ for all $1 \leq i < k$, and Q is maximal, define the extended dead region $\mathcal{D}_1(Q)$ with respect to the red root r_1 to be:

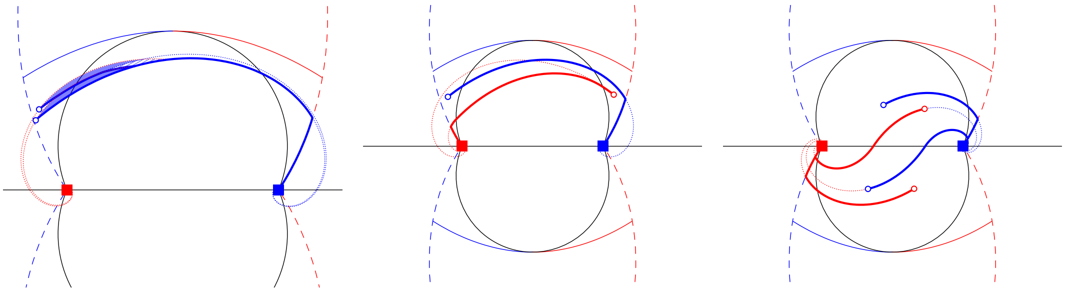
$$\mathcal{D}_1(Q) = \bigcup_{1 \leq i < k} F_i \,,$$

where F_i is the area enclosed between the left sides of $\mathcal{R}(q_i, r_2)$ and $\mathcal{R}(q_{i+1}, r_2)$, and the two red spiral segments $s_r^+(q_i)$ and $s_r^+(q_k)$.

Similarly define the extended dead region $\mathcal{D}_1(Q)$ of a set Q of blue terminals lying in the bottom half-plane, and the extended dead regions $\mathcal{D}_2(P)$ of a set P of red terminals for the top and the bottom half-planes.

We will show that there exists a non-crossing drawing of two flux trees, given that the roots are not contained in any $\mathcal{R}$-region, if and only if no terminal lies inside a dead region or an extended dead region of the other color.

Routing rules. We can partition $\mathbb{R}^2$ into several regions such that we can specify the routing rules for terminals within each region separately (see Figure 13): A^t and A^b are bounded by r_1r_2 and two circular arcs with an angle subtended by the chord r_1r_2 equal to $\pi - 2\alpha$; B_1^t, B_2^t, B_1^b, and B_2^b are bounded by the arcs of A^t and A^b, the boundaries of the regions $\mathcal{C}_r$ and $\mathcal{C}_b$, and by a spiral going through the topmost or the bottommost point of the arcs; $D^t = \mathbb{R}^2_+ \backslash (\mathcal{C}_r \cup \mathcal{C}_b \cup A^t \cup B_1^t \cup B_2^t)$, and $D^b = \mathbb{R}^2_- \backslash (\mathcal{C}_r \cup \mathcal{C}_b \cup A^b \cup B_1^b \cup B_2^b)$. An important observation is that in A^t and A^b red paths can be routed along blue spirals and vice versa.



(a) Rule 2. **(b)** Rule 4. **(c)** Rules 4 and 5.

Figure 14 Routing rules for drawing flux trees.

We can now introduce the following routing rules to draw two non-crossing flux trees if there were no terminals of the other color in the dead regions. Refer to Figure 14 for the illustrations of the rules. Note that Rules 4 and 5 can introduce some intersections, that we will uncross afterwards.

Rule 1. Route red and blue terminals in D^t in D^b along their respective spiral segments s_r^+, s_b^-, s_r^-, or s_b^+ until reaching the boundary of $\mathcal{C}_r$ or $\mathcal{C}_b$;

Rule 2. For all blue extended dead regions, route the corresponding sets of blue terminals $Q = \{q_1, \dots, q_k\}$ in $\mathbb{R}^2_+$ in the following way: route q_1 along a blue spiral segment $s_b^-(q_1)$ until it reaches $\mathcal{C}_b$; for every $1 < i \leq k$, route q_i along the boundary of its dead region and then along the boundary of the extended dead region until reaching $s_b^-(q_1)$ (merging with the path from q_1 to r_2) or the boundary of the cone $\mathcal{C}_b$;

Rule 3. For all red extended dead regions, route the corresponding sets of red terminals $P = \{p_1, \dots, p_k\}$ in $\mathbb{R}^2_-$ in the following way: route p_1 along a red spiral segment $s_r^-(p_1)$ until it reaches $\mathcal{C}_r$; for every $1 < i \leq k$, route p_i along the boundary of its dead region, then along the boundary of the extended dead region until reaching $s_r^-(p_1)$ (merging with the path from p_1 to r_1) or the boundary of the cone $\mathcal{C}_r$;

Rule 4. The rest of the blue terminals in $\mathbb{R}^2_+$ route along their left blue spiral segments until reaching the boundary of cone $\mathcal{C}_b$; the rest of the red terminals in $\mathbb{R}^2_+$ route along a right red spiral within region B_2^t, along a left blue spiral within A^t, and along right red spiral within B_1^t until reaching the cone $\mathcal{C}_r$;

Rule 5. The rest of the red terminals in $\mathbb{R}^2_-$ route along their left red spiral segments until reaching the boundary of cone $\mathcal{C}_r$; the rest of the blue terminals in $\mathbb{R}^2_-$ route along a right blue spiral within B_1^b, along a left red spiral within A^b, and along right blue spiral within B_2^b until reaching the cone $\mathcal{C}_b$;

Rule 6. Finally, route all the blue paths along the boundary of $\mathcal{C}_b$ to r_2 and all the red paths along the boundary of $\mathcal{C}_r$ to r_1. Red terminals in $\mathcal{C}_r$ and blue terminals in $\mathcal{C}_b$ can be routed arbitrarily (joining when necessary) within those cones towards their respective roots.

As mentioned, after applying Rules 4 and 5, some intersections are possible. Specifically, red and blue paths can intersect within regions B_1^t, B_2^t, B_1^b, or B_2^b. Red and blue paths do not intersect within A^t or A^b, as they follow non-intersecting spirals; and red and blue paths do not intersect within D^t or D^b, otherwise the corresponding $\mathcal{R}$-regions would fully intersect.

Consider a red terminal p and a blue terminal q in $\mathbb{R}^2_+$ such that their paths, constructed by the presented routing rules, intersect. These paths can intersect only once, due to the

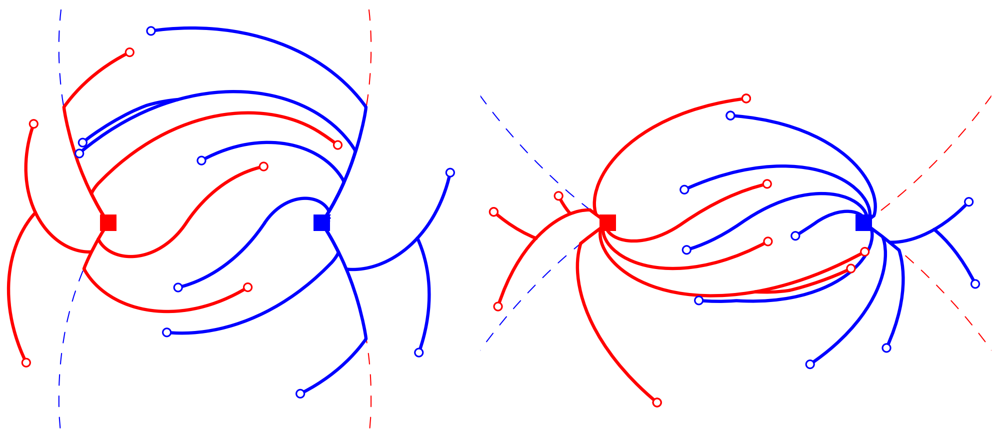


Figure 15 Two non-crossing drawings of flux trees for $\alpha = 60°$ (left) and $\alpha = 30°$ (right).

difference of curvatures of spirals in B_1^t and B_2^t, and because within A^t these paths follow non-intersecting spirals. There can be two cases: (a) the intersection point is in B_1^t; and (b) the intersection point is in B_2^t.

In the first case, the blue terminal q is in B_1^t, and the red path crosses its dead region $\mathcal{D}_1(q)$. Then reroute the red path along the boundary of the dead region $\mathcal{D}_1(q)$, when it first encounters it. If q is a part of a set of blue terminals that define an extended dead region, then reroute the red path along the boundary of the extended dead region when it first encounters it. The new red path will not intersect any other blue paths, otherwise these paths would be a part of the set defining the extended dead region.

In the second case, when intersection point is inside B_2^t, the red terminal p is in B_2^t. Let f be the intersection point of the red path with the boundary between A^t and B_2^t. Consider the left side of $\mathcal{R}(p, r_q)$. It intersects the blue path exactly two times, otherwise the $\mathcal{R}$-regions of p and q would fully intersect. Let g be the intersection point of the left side of $\mathcal{R}(p, r_q)$ and the blue path inside A^t. Consider all the blue paths that intersect the red spiral segment $s_r^+(p)$ between points f and g. Reroute all these paths along the red spiral segment $s_r^+(p)$ when they first encounter it, until they reach point g, then route the merged path along a new blue spiral segment $s_b^-(g)$ within B_2^t. Let the red path continue following the red spiral segment $s_r^+(p)$ when it enters A^t until it reaches point f, then let the red path follow the blue spiral segment $s_b^-(f)$ "parallel" to the rest of the paths in A^t. Note, that if there was a part of the red path in $\mathbb{R}^2_-$, the new path may completely lie in $\mathbb{R}^2_+$. This procedure essentially brings the part of the blue path(s) that was above the red path under it.

The symmetrical cases in the bottom half-plane can be dealt with similarly. And if the terminals lie in the different half-planes, their paths can intersect once or twice. However, the method for uncrossing such paths completely mirrors the cases for when p and q lie in the same half-plane. Figure 15 shows the final result of the procedure. In the full version of this paper we prove the following theorem.

▶ **Theorem 10.** *A drawing of two non-crossing flux trees with no root containment in $\mathcal{R}$-regions exists if and only if no terminal lies in a dead region or an extended dead region of the other color. If it exists, such a drawing can be constructed in polynomial time.*

4 Conclusion and Future Work

In this paper we study the problem of drawing a flow map with non-crossing curves that have to be oriented approximately towards the source. We have shown that we can efficiently decide if two rectilinear Steiner arborescences can be drawn without crossings, if we require the paths to simply be xy-monotone with no other restrictions. Similarly, we show how to draw two non-intersecting flux trees in the case when their roots are not contained in the other tree's $\mathcal{R}$-regions.

With an extra restriction on the paths that prohibits free turns, the problem becomes NP-hard for k Steiner arborescences, where k is part of the input. We conjecture that this problem is also NP-hard for $k = 2$. Whether the problem is NP-hard for more than two Steiner arborescences in the free turn model is left as an open problem.

References

1. Oswin Aichholzer, Franz Aurenhammer, Thomas Hackl, and Clemens Huemer. Connecting colored point sets. *Discrete Applied Mathematics*, 155(3):271–278, 2007. `doi:10.1016/j.dam.2006.06.010`.
2. Oswin Aichholzer, Franz Aurenhammer, Christian Icking, Rolf Klein, Elmar Langetepe, and Günter Rote. Generalized self-approaching curves. *Discrete Applied Mathematics*, 109(1-2):3–24, apr 2001. `doi:10.1016/S0166-218X(00)00233-X`.
3. Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.
4. Sergey Bereg, Krzysztof Fleszar, Philipp Kindermann, Sergey Pupyrev, Joachim Spoerhase, and Alexander Wolff. Colored non-crossing Euclidean Steiner forest. In *Proc. 26th International Symposium on Algorithms and Computation (ISAAC)*, pages 429–441, 2015. URL: `http://arxiv.org/abs/1509.05681`.
5. Kevin Buchin, Bettina Speckmann, and Kevin Verbeek. Angle-restricted Steiner arborescences for flow map layout. *Algorithmica*, 72(2):656–685, 2015. `doi:10.1007/s00453-013-9867-z`.
6. Klaus Jansen and Gerhard J. Woeginger. The complexity of detecting crossingfree configurations in the plane. *BIT Numerical Mathematics*, 33(4):580–595, 1993. `doi:10.1007/BF01990536`.
7. 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):301–306, 2005.
8. Christian Knauer, Étienne Schramm, Andreas Spillner, and Alexander Wolff. Configurations with few crossings in topological graphs. *Computational Geometry*, 37(2):104–114, 2007. `doi:10.1016/j.comgeo.2006.06.001`.
9. J. Leaños, C. Merino, G. Salazar, and J. Urrutia. Spanning trees of multicoloured point sets with few intersections. In *Proc. Indonesia-Japan Joint Conference on Combinatorial Geometry and Graph Theory (IJCCGGT)*, LNCS 3330, pages 113–122, 2005.
10. Bing Lu and Lu Ruan. Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. *Journal of Combinatorial Optimization*, 4(3):357–363, 2000. `doi:10.1023/A:1009826311973`.
11. Julia Schüler and Andreas Spillner. Crossing-free spanning trees in visibility graphs of points between monotone polygonal obstacles. In *Proc. 9th International Computer Science Symposium in Russia (CSR)*, LNCS 8476, pages 337–350, 2014.
12. Shin-ichi Tokunaga. Intersection number of two connected geometric graphs. *Information Processing Letters*, 59(6):331–333, 1996. `doi:10.1016/0020-0190(96)00124-X`.