Angle-Restricted Steiner Arborescences for Flow Map Layout*

Kevin Buchin† Bettina Speckmann† Kevin Verbeek†

Abstract

Flow maps visualize the movement of objects between places. One or more sources are connected to several targets by arcs whose thickness corresponds to the amount of flow between a source and a target. Flow maps reduce visual clutter by merging (bundling) lines smoothly and by avoiding self-intersections.

We present algorithms that compute crossing-free flows of high visual quality. To this end we introduce a new variant of the geometric Steiner arborescence problem. The goal is to connect the targets to a source with a tree of minimal length whose arcs obey a certain restriction on the angle they form with the source. Such an *angle-restricted Steiner arborescence*, or simply *flow tree*, naturally induces a clustering on the targets and smoothly bundles arcs.

We study the properties of optimal flow trees and show that they consist of logarithmic spirals and straight lines. Computing optimal flow trees is NP-hard. Hence we consider a variant of flow trees which uses only logarithmic spirals, so called *spiral trees*. Spiral trees approximate flow trees within a factor depending on the angle restriction. Computing optimal spiral trees remains NP-hard. We present an efficient 2-approximation for spiral trees, which can be extended to avoid certain types of obstacles.

1 Introduction

Flow maps are a cartographic method for the visualization of the movement of objects between places [9]. One or more sources are connected to several targets by arcs whose thickness corresponds to the amount of flow between a source and a target. Most flow maps are drawn by hand and few automated methods exist.

Good flow maps share some common properties. They reduce visual clutter by merging (bundling) lines as smoothly and frequently as possible. Furthermore, they strive to avoid crossings between lines. Specifically, single-source flows are drawn entirely without crossings. Flow maps that depict trade often route edges along actual shipping routes. In that case a moderate distortion of the underlying geography is admissible. In contrast, flow maps that depict more abstract data, such as migration or internet traffic, do not distort the geography of the underlying map. In addition, flow maps often try to avoid covering important map features with flows to aid recognizability.

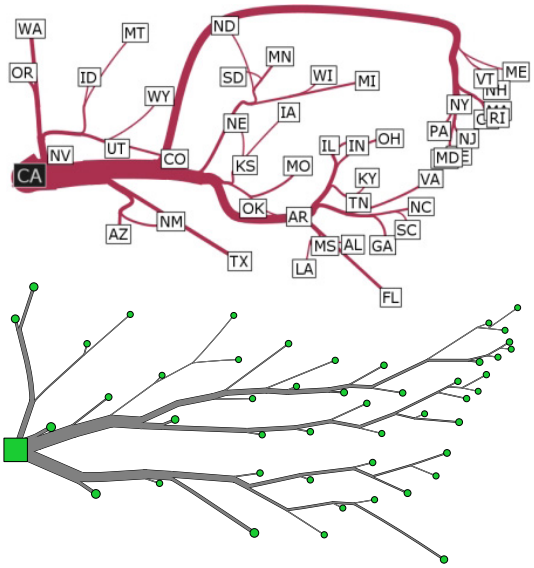

Figure 1: Two maps based on the same data set: migration from California 1995–2000. Phan *et al.* [4] (top), and the output of our algorithm (bottom).

Modeling and formal problem description. We study the problem of computing crossing-free single-source flows of high visual quality. Our input consists of a point r, the *root* or *source*, and n points $t_1, \ldots, t_n$, the *terminals* or *targets*. For every target t_i, we are also given a weight w_i, representing the amount of flow from the source to target t_i. Visually appealing flows merge quickly, but smoothly. Disregarding weights, a geometric minimal *Steiner arborescence* on our input results in the shortest possible tree, which naturally merges quickly. A Steiner arborescence for a given root and a set of terminals is a rooted directed *Steiner tree*, which contains all terminals and where all edges are directed away from the root. Without additional restrictions on the edge directions (as in the rectilinear case or in the variant proposed below), a geometric Steiner arborescence is simply a geometric Steiner tree with directed edges. Steiner arborescences have angles of $2\pi/3$ at every internal node and hence do not have the smooth appearance of hand-drawn flow maps. This motivates us to introduce a new variant of the geometric minimal Steiner arborescence problem.

*B. Speckmann and K. Verbeek are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

†Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands, k.a.buchin@tue.nl, speckman@win.tue.nl, and k.a.b.verbeek@tue.nl

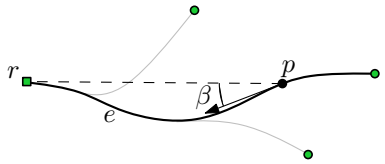

Figure 2: The angle restriction.

The goal is to connect the terminals to the root with a tree of minimal length whose arcs obey a certain restriction on the angle they form with the root.

Specifically, we use a *restricting angle* $\alpha < \pi/2$ to control the direction of the arcs of a Steiner arborescence T. Consider a point p on an arc e from a terminal to the root (see Fig. 2). Let β be the angle between the vector from p to the root r and the tangent vector of e at p. We require that $\beta \leq \alpha$ for all points p on T. We call a Steiner arborescence which obeys this angle restriction an *angle-restricted Steiner arborescence*, or simply *flow tree*. Note, that this definition is not taking weights into account. Hence an optimal flow tree is simply an angle-restricted Steiner arborescence of minimal length. Here and in the remainder of the paper it is convenient to direct flow trees from the terminals to the root. Also, to simplify descriptions, we often identify the nodes of a flow tree T with their locations in the plane.

In the context of flow maps it is important that flow trees can avoid obstacles, which model important features of the underlying geographic map. It is also undesirable that terminals become internal nodes of a flow tree. We can avoid this by covering each terminal with an obstacle. Hence our input also includes a set of m obstacles $B_1, \ldots, B_m$. We denote the total complexity of all obstacles by M. In the presence of obstacles our goal is to find the shortest flow tree T that is planar and avoids the obstacles. Finally, most flow maps use thick edges to indicate the amount of flow. To this end we consider *weighted flow trees*. The arcs of a weighted flow tree T must satisfy the following conditions [9]. If an arc e starts at a terminal t_i then the thickness of e must be proportional to w_i. If an arc e starts at an internal node u of T then its thickness must be proportional to the sum of the thicknesses of the arcs terminating in u. We require, of course, that thick arcs do not overlap obstacles.

Related work. One of the first systems for the automated creation of flow maps was developed by Tobler in the 1980s [10, 11]. His system does not use edge bundling and hence the resulting maps suffer from visual clutter. A couple of years ago Phan *et al.* [4] presented an algorithm, based on hierarchical clustering of the terminals, which creates single-source flow maps with bundled edges. This algorithm uses an iterative ad-hoc method to route edges and hence is often unable to avoid crossings.

There are many variations on the classic Steiner tree problem which use metrics that are related to their specific applications. Of particular relevance is the *rectilinear Steiner arborescence* (RSA) problem. Here we are given a root at the origin and a set of terminals $t_1, \ldots, t_n$ in the northeast quadrant of the plane. The goal is to find the shortest rooted rectilinear tree T with all edges directed away from the root, such that T contains all points $t_1, \ldots, t_n$. For any edge T from $p = (x_p, y_p)$ to $q = (x_q, y_q)$ it must hold that $x_p \leq x_q$ and $y_p \leq y_q$. If we drop the condition of rectilinearity then we arrive at the *Euclidean Steiner arborescence* (ESA) problem. In both cases it is NP-hard [7, 8] to compute a tree of minimum length. Rao *et al.* [6] give a simple 2-approximation algorithm for minimum rectilinear Steiner arborescences. Córdova and Lee [1] describe an efficient heuristic which works for terminals located anywhere in the plane. Ramnath [5] presents a more involved 2-approximation that can also deal with rectangular obstacles. Finally, Lu and Ruan [2] developed a PTAS for minimum rectilinear Steiner arborescences.

Results and organization. Section 2 gives properties of optimal flow trees. In particular, the arcs of optimal flow trees consist of (segments of) logarithmic spirals and straight lines. Flow trees naturally induce a clustering on the terminals and smoothly bundle lines. In the full paper we show that it is NP-hard to compute optimal flow trees. Hence, Section 3 introduces a variant of flow trees, so called *spiral trees*. The arcs of spiral trees consist only of logarithmic spiral segments. We prove that spiral trees approximate flow trees within a factor depending on the restricting angle α. Computing optimal spiral trees remains NP-hard. But we can establish a connection between spiral trees and rectilinear Steiner arborescences. Based on that in Section 4 we develop a 2-approximation algorithm for spiral trees which runs in $O(n \log n)$ time. In the full paper we extend our approximation algorithm to include certain types of obstacles. Finally, in Section 5 we briefly comment on weighted flow trees.

2 Optimal flow trees

Recall that our input consists of a root r, terminals $t_1, \ldots, t_n$, and a restricting angle $\alpha < \pi/2$. We can assume that the root lies at the origin. Recall further that an optimal flow tree is a geometric Steiner arborescence, whose arcs are directed from the terminals to the root and that satisfies the angle restriction.

Spiral regions. For a point p in the plane, we consider the region $\mathcal{R}_p$ of all points that are *reachable* from p with an *angle-restricted* path, that is, with a path that satisfies the angle restriction. Clearly, the root r is always in $\mathcal{R}_p$. The boundaries of $\mathcal{R}_p$ consist of curves that follow one of the two directions that form exactly an angle α with the direction towards

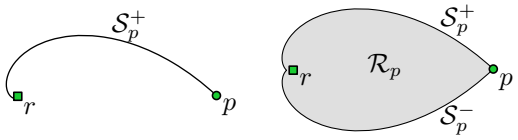

Figure 3: Spirals and spiral regions.

the root. Curves with this property are known as *logarithmic spirals* (see Fig. 3). Logarithmic spirals are self-similar; scaling a logarithmic spiral results in another logarithmic spiral. As all spirals in this paper are logarithmic, we simply refer to them as *spirals*. For $\alpha < \pi/2$ there are two spirals through a point. The *right spiral* $\mathcal{S}_p^+$ is given by the following parametric equation in polar coordinates, where $p = (R, \phi)$: $R(t) = Re^{-t}$ and $\phi(t) = \phi + \tan(\alpha)t$. The parametric equation of the *left spiral* $\mathcal{S}_p^-$ is the same with α replaced by $-\alpha$. Note that a right spiral $\mathcal{S}_p^+$ can never cross another right spiral $\mathcal{S}_q^+$ (the same holds for left spirals). The spirals $\mathcal{S}_p^+$ and $\mathcal{S}_p^-$ cross infinitely often. The reachable region $\mathcal{R}_p$ is bounded by the parts of $\mathcal{S}_p^+$ and $\mathcal{S}_p^-$ with $0 \leq t \leq \pi \cot(\alpha)$. We therefore call $\mathcal{R}_p$ the *spiral region* of p. It follows directly from the definition that for all $q \in \mathcal{R}_p$ we have that $\mathcal{R}_q \subseteq \mathcal{R}_p$.

Lemma 1 *The shortest angle-restricted path between a point p and a point $q \in \mathcal{R}_p$ consists of a straight segment followed by a spiral segment. Either segment can have length zero.*

Using Lemma 1 we establish the following properties.

Property 1 *An optimal flow tree consists of straight segments and spiral segments (see Fig. 4).*

Property 2 *Every node in an optimal flow tree, other than the root, has at most two incoming edges.*

Property 3 *Every optimal flow tree is planar.*

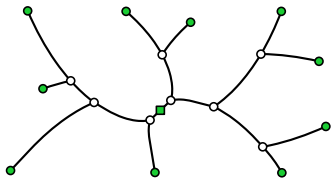

Figure 4: An optimal flow tree ($\alpha = \pi/6$).

3 Spiral trees

In this section we introduce *spiral trees* and prove that they approximate flow trees. The arcs of a spiral tree consist only of spiral segments of a given α. An optimal spiral tree is hence the shortest flow tree that uses only spiral segments. Any particular arc of a spiral tree can consist of arbitrarily many spiral segments; it can switch between following its right spiral and following its left spiral an arbitrary number of times. The length of a spiral segment can be expressed in polar coordinates. Let $p = (R_1, \phi_1)$ and $q = (R_2, \phi_2)$ be two points on a spiral, then the distance $D(p, q)$ between p and q on the spiral is

$$D(p, q) = \sec(\alpha)|R_1 - R_2|\,. \tag{1}$$

Consider now the shortest *spiral path*—using only spiral segments—between a point p and a point q reachable from p. The reachable region for p is still its spiral region $\mathcal{R}_p$, so necessarily $q \in \mathcal{R}_p$. The length of a shortest spiral path is given by Equation 1. The shortest spiral path is not unique, in particular, any sequence of spiral segments from p to q is shortest, as long as we keep moving towards the root.

Theorem 2 *The optimal spiral tree T' is a $\sec(\alpha)$-approximation of the optimal flow tree T.*

Proof. Let $\mathcal{C}_R$ be a circle of radius R with the root r as center. A lower bound for the length of T is given by $L(T) \geq \int_0^\infty |T \cap \mathcal{C}_R| dR$, where $|T \cap \mathcal{C}_R|$ counts the number of intersections between the tree T and the circle $\mathcal{C}_R$. Using Equation 1, the length of T' is $L(T') = \sec(\alpha) \int_0^\infty |T' \cap \mathcal{C}_R| dR$. Now consider the spiral tree T'' with the same nodes as T, but where all arcs between the nodes are replaced by a sequence of spiral segments (see Fig. 5). For a given circle $\mathcal{C}_R$, this operation does not change the number of intersections of the tree with $\mathcal{C}_R$, i.e. $|T \cap \mathcal{C}_R| = |T'' \cap \mathcal{C}_R|$. So we have that $L(T') \leq L(T'') \leq \sec(\alpha) L(T)$. □

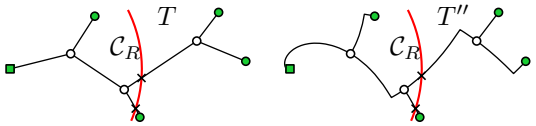

Figure 5: T and T''.

Observation 1 *An optimal spiral tree is planar and every node $u \neq r$ has at most two incoming edges.*

Relation with rectilinear Steiner arborescences. Both rectilinear Steiner arborescences and spiral trees contain directed paths, from the root to the terminals or vice versa. The edges of a rectilinear Steiner arborescence are restricted to point right or up, which is similar to the angle restriction of spiral trees. The relation between the concepts cannot be used directly, but some of the techniques developed for rectilinear Steiner arborescences can be adapted to spiral trees.

4 Computing spiral trees

In the special case that the spiral regions of the terminals are empty, i.e., if $t_i \notin \mathcal{R}_{t_j}$ for all $i \neq j$, we can use dynamic programming to compute an optimal spiral tree in $O(n^3)$ time, based on the following lemma.

Lemma 3 *If the spiral regions of all terminals are empty, then the leaf order of any planar spiral tree follows the radial order of the terminals.*

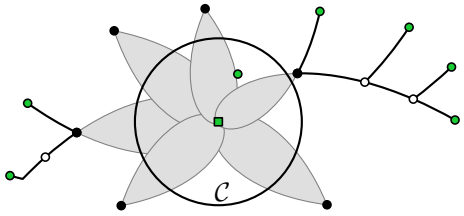

Figure 6: The wavefront $\mathcal{W}$.

We now describe an approximation algorithm that is based on a greedy algorithm for rectilinear Steiner arborescences [6]. We iteratively join nodes, possibly with Steiner nodes, until all terminals are connected in a single tree T, the *greedy spiral tree*. Initially, T is a forest. A node (or terminal) is *active* if it does not have a parent in T. In every step, we join the two active nodes for which the *join point* is farthest from r. The join point p_{uv} of two nodes u and v is the farthest point p from r such that $p \in \mathcal{R}_u \cap \mathcal{R}_v$. This point is unique if u, v, and r are not collinear.

The algorithms sweeps a circle $\mathcal{C}$, centered at r, inwards over all terminals. All active nodes that lie outside of $\mathcal{C}$ form the *wavefront* $\mathcal{W}$ (the black nodes in Fig. 6). $\mathcal{W}$ is implemented as a balanced binary search tree, where nodes are sorted according to the radial order around r. We join two active nodes u and v as soon as $\mathcal{C}$ passes over p_{uv}. For any two nodes $u, v \in \mathcal{W}$ it holds that $u \notin \mathcal{R}_v$. Hence, when $\mathcal{C}$ passes over p_{uv} and both nodes u and v are still active, then, by Lemma 3, u and v must be neighbors in $\mathcal{W}$. We process the following events.

Terminal. When $\mathcal{C}$ reaches a terminal t, we add t to $\mathcal{W}$. We need to check if there exists a neighbor v of t in $\mathcal{W}$ such that $t \in \mathcal{R}_v$. If such a node v exists, then we remove v from $\mathcal{W}$ and connect v to t. Finally we compute new join point events for t and its neighbors in $\mathcal{W}$.

Join point. When $\mathcal{C}$ reaches a join point p_{uv} (and u and v are still active), we connect u and v to p_{uv}. Next, we remove u and v from $\mathcal{W}$ and we add p_{uv} to $\mathcal{W}$ as a Steiner node. Finally we compute new join point events for p_{uv} and its neighbors in $\mathcal{W}$.

We store the events in a priority queue $\mathcal{Q}$, ordered by decreasing distance to r. It is easy to verify that the total number of events is $O(n)$, and that each event can be handled in $O(\log n)$ time, so the total running time is $O(n \log n)$. The greedy spiral tree is planar by construction. We can prove the following results.

Lemma 4 *Let $\mathcal{C}$ be any circle centered at r and let T and T' be the optimal spiral tree and the greedy spiral tree, respectively. Then $|\mathcal{C} \cap T'| \leq 2|\mathcal{C} \cap T|$.*

Theorem 5 *The greedy spiral tree is a 2-approximation of the optimal spiral tree and can be computed in $O(n \log n)$ time.*

5 Weighted flow trees

When creating flow maps we need to consider weighted flow trees, that is, flow trees with thick arcs. To facilitate easy comparisons between flows these arcs should be drawn as thickly as possible. However, increasing the thickness can increase the length of the optimal flow tree substantially. This tradeoff between arc thickness and tree length makes it very difficult to produce theoretically interesting results regarding weighted flow trees. Hence we implemented a heuristic approach based on our approximation algorithm for (thin) spiral trees. We thicken the edges of the greedy spiral tree, pushing arcs away from terminals and obstacles, and apply local changes to the tree topology to facilitate thicker flows. The resulting flow maps are still guaranteed to be crossing free. First results look very promising (see Fig. 1); our maps are less cluttered than those produced by other automated methods, and we can observe that spiral trees naturally cluster terminals well.

References

[1] J. Córdova and Y. Lee. A Heuristic Algorithm for the Rectilinear Steiner Arborescence Problem. Technical Report, Engineering Optimization, 1994.

[2] B. Lu and L. Ruan. Polynomial Time Approximation Scheme for the Rectilinear Steiner Arborescence Problem. *Journal of Combinatorial Optimization*, 4(3):357–363, 2000.

[3] J. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8(1):55–88, 1992.

[4] D. Phan, L. Xiao, R. Yeh, P. Hanrahan and T. Winograd. Flow map layout. In *Proc. IEEE Symposium on Information Visualization*, pp. 219–224, 2005.

[5] S. Ramnath. New approximations for the rectilinear Steiner arborescence problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(7):859–869, 2003.

[6] S.K. Rao, P. Sadayappan, F.K. Hwang and P.W. Shor. The rectilinear Steiner arborescence problem. *Algorithmica*, 7(1):277–288, 1992.

[7] W. Shi and C. Su. The Rectilinear Steiner Arborescence Problem Is NP-Complete. In *Proc. 11th ACM-SIAM Symposium on Discrete Algorithms*, pp. 780–787, 2000.

[8] W. Shi and C. Su. The Rectilinear Steiner Arborescence Problem Is NP-Complete. *SIAM Journal on Computing*, 35(3):729–740, 2005.

[9] T.A. Slocum, R.B. McMaster, F.C. Kessler and H.H. Howard. Thematic Cartography and Geovisualization. Pearson, New Jersey, 2010.

[10] *CSISS - Spatial Tools: Tobler's Flow Mapper.* http://www.csiss.org/clearinghouse/FlowMapper/.

[11] W. Tobler. Experiments in Migration Mapping by Computer. *The American Cartographer*, 14(2):155–163, 1987.