# Area-Proportional Drawings of Intersecting Families of Simple Closed Curves

Stirling Chow* Frank Ruskey†

**Abstract**

A FISC, or *family of intersecting simple closed curves*, is a collection of simple closed curves in the plane with the properties that there is some open region common to the interiors of all the curves, and that every two curves intersect in finitely many points or arcs.

Let $\mathcal{F}$ be a FISC with a set of open regions $R$. $\mathcal{F}$ is said to be *area-proportional* with respect to weight function $\omega : R \rightarrow \mathbb{R}^+$ if there is a positive constant $\alpha$ such that for any two finite regions, $r_1$ and $r_2$, $area(r_1)/area(r_2) = \alpha\omega(r_1)/\omega(r_2)$.

We consider $\mathcal{F}$ as a directed plane graph, $\vec{G}(\mathcal{F})$, where the curve intersections are vertices and the curve arcs between vertices are edges. Edges are directed so that each of $\mathcal{F}$'s curves is traversed in a clockwise fashion. The directed plane dual of $\vec{G}(\mathcal{F})$, denoted $\vec{D}(\mathcal{F})$, has edges oriented to indicate inclusion in fewer interiors of the curves. The graph $\vec{G}(\mathcal{F})$ has an area-proportional drawing with respect to $\omega$ if there is some FISC $\mathcal{C}$ that is area-proportional to $\omega$ and where $\mathcal{F}$ can be transformed into $\mathcal{C}$ by a continuous transformation of the plane. We describe an $O(n|V|)$ algorithm for creating an area-proportional drawing of $\vec{G}(\mathcal{F}) = (V, E)$ where $\mathcal{F}$ is a FISC with $n$ curves and $\vec{D}(\mathcal{F})$ has only one source and only one sink. For the case of $n$-Venn diagrams, since $|V| \leq 2^n - 2$, this yields an $O(|V|lg|V|)$ drawing algorithm.

## 1 Introduction

A drawing of a plane graph $G = (V, E, F)$ is said to be *area-proportional* with respect to weight function $\omega : F \rightarrow \mathbb{R}^+$ if there is a positive constant $\alpha$ such that for any two finite faces, $f_1$ and $f_2$, $area(f_1)/area(f_2) = \alpha\omega(f_1)/\omega(f_2)$.

*Cartograms* [10] are a common application of area-proportional drawings whereby the regions of a geographic map are distorted to represent quantitative values (e.g., population). *Euler and Venn diagrams* [8, 7] are a visual representation of a set system; in such diagrams, each class of objects is represented by a single curve, and the regions of overlap represent objects that belong to more than one class. Area-proportionality has been used to augment Euler and Venn diagrams with a perceptual indication of the set sizes [1, 4].

In this paper, we present an algorithm for creating area-proportional drawings of a special class of plane graphs that represent *families of intersecting simple closed curves*, or FISCs. A FISC $\mathcal{F}$ is a collection of simple closed curves in the plane with the properties that there is some open region common to the interiors of all curves, and that every two curves intersect in finitely many points or arcs. We consider $\mathcal{F}$ as a directed plane graph, $\vec{G}(\mathcal{F})$, where the curve intersections are vertices and the curve arcs between vertices are edges. Edges are directed so that each of $\mathcal{F}$'s curves is traversed in a clockwise fashion. The directed plane dual of $\vec{G}(\mathcal{F})$, denoted $\vec{D}(\mathcal{F})$, has edges oriented to indicate inclusion in fewer interiors of the curves. Figure 1 shows an example of a FISC and its associated graphs.

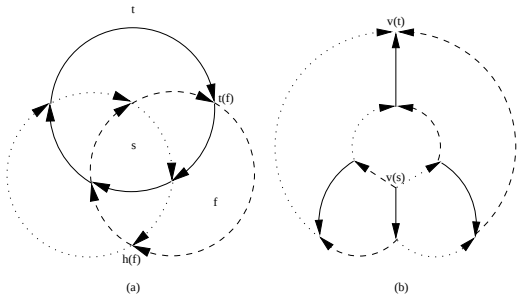


Figure 1: (a) A three curve FISC $\mathcal{F}$ represented by a directed plane graph $\vec{G}(\mathcal{F})$. (b) The corresponding directed dual $\vec{D}(\mathcal{F})$.

Our algorithm is an extension of previous work by Bultena, Grünbaum, and Ruskey [3], that considered drawings of FISCs where each curve was convex. The proofs of their results involved continuous transformations of the plane; in addition to drawing area-proportional FISCs, our algorithm provides a discrete combinatorial proof of their results.

We conclude by relating our algorithm to existing cartogram algorithms and consider how it can be applied to generate and draw area-proportional Euler and Venn diagrams.

*Department of Computer Science, University of Victoria, Canada, schow@cs.uvic.ca. Research supported by an NSERC CGS.

†Department of Computer Science, University of Victoria, Canada. Research supported in part by NSERC.

## 2 Mathematical Background

Let $\vec{G}(\mathcal{F})$ be a directed plane graph representing FISC $\mathcal{F}$. Each face $f$ of $\vec{G}(\mathcal{F})$ is associated with a vertex $v(f)$ in the directed dual $\vec{D}(\mathcal{F})$.

If $\vec{D}(\mathcal{F})$ has a single source $v(s)$ and a single sink $v(t)$, then $\mathcal{F}$ is said to be *monotone*. A FISC is *ray-monotone* if there exists a point $x$ such that every ray emanating from $x$ intersects each curve exactly once. The following lemma by Bultena, Grünbaum, and Ruskey [3] relates monotonicity and ray-monotonicity.

**Lemma 1** *If a FISC is monotone, then it is isomorphic (by a continuous transformation of the plane) to a ray-monotone FISC.*

The proof of Lemma 1 involves continuous transformations of the plane along laminar flows and does not provide a discrete algorithm for defining the isomorphism. One of the consequences of our algorithm is a discrete combinatorial proof of Lemma 1 based on the following lemma.

**Lemma 2** *Let $P$ be a path in $\vec{D}(\mathcal{F})$ from $v(s)$ to $v(t)$. The graph that results from removing $P$'s respective dual edges from $\vec{G}(\mathcal{F})$ is acyclic.*

**Proof.** Omitted for brevity. □

A consequence of monotonicity is that each face of $\vec{G}(\mathcal{F})$ is comprised of two directed paths that share a common source and a common sink (see Fig. 1(a)), as described in the following lemma.

**Lemma 3** *If $\mathcal{F}$ is monotone (i.e., $\vec{D}(\mathcal{F})$ has a single source $v(s)$ and a single sink $v(t)$), then the perimeter of any face $f$ of $\vec{G}(\mathcal{F})$, other than $s$ and $t$, is comprised of two directed paths that share a common head $h(f)$ and a common tail $t(f)$.*

**Proof.** Consider the proof of Lemma 1.1 [3] in terms of $\vec{G}(\mathcal{F})$ rather than $\vec{D}(\mathcal{F})$. □

## 3 Algorithm

Based on Lemmas 1–3, our algorithm draws $\vec{G}(\mathcal{F})$ so that it is area-proportional with respect to a weight function $\omega$. The algorithm is comprised of the following major steps:

1. Find a path in $\vec{D}(\mathcal{F})$ between $v(s)$ and $v(t)$ and remove the respective dual edges from $\vec{G}(\mathcal{F})$ to create an acyclic graph $\vec{G}'(\mathcal{F})$.
2. Topologically order the vertices of $\vec{G}'(\mathcal{F})$ and associate each vertex with a ray; order the rays clockwise about an origin $x$ to match the topological order.
3. For the common region $s$ of $\mathcal{F}$, draw a polygon with area $\omega(s)$ that contains point $x$.
4. For each vertex $v(f)$ of $\vec{D}(\mathcal{F})$ in breadth-first order beginning at $v(s)$, draw face $f$ as a polygon with area $\omega(f)$ by expanding one of $f$'s directed paths outward from $x$ along the rays between $h(f)$ and $t(f)$.

Steps 1 and 2 compute a transformation that defines the isomorphism between a monotone FISC and a ray-monotone FISC according to Lemma 1.

Step 3 draws $\vec{G}(\mathcal{F})$'s common face.

Step 4 draws the remaining faces outwards from the common face along the rays defined by the topological sort in Step 2. Because of Lemma 3, as each face $f$ is encountered, one of its paths will be *fixed* (i.e., already defined by the previously drawn polygons) and the other will be *free* to expand outward within the sector defined by the rays associated with $h(f)$ and $t(f)$. The ray-monotoncity ensures that as each face expands outward, it will not collide with any faces that have already been drawn.

Algorithm 1 elaborates on the details and is the basis for the complexity analysis in the next section. To make the diagram more compact, Alg. 1 layers the independent vertices of each step of the topological sort on a single ray. In addition, Alg. 1 inserts a dummy ray between each topological ray so that faces whose free path has only a single edge are able to expand (since $h(f)$ and $h(t)$ are fixed).

The details of how to compute the polygons in Alg. 1 have been intentionally omitted to emphasize the flexibility of the algorithm to produce a wide-range of drawings of $\vec{G}(\mathcal{F})$. There are many choices for the center polygon in Line 26; for example, a regular $2k$-gon will yield rays that are evenly distributed. For subsequent polygons, there is a choice as to how the free path expands outwards from the fixed path; for example, the choice could be made to have the free path expand uniformly from the fixed path or as tangentially to the rays as possible (which minimizes jaggedness). In all of these cases, the relevant formulae are easily derived using basic geometry and trigonometry. Figure 2 shows two drawings the FISC from Fig. 1(a) where each region has equal area; the faces are numbered according to the order in which they were drawn. Also, in this implementation, two dummy rays are added between each vertex ray (to provide more smoothness).

## 4 Analysis

**Theorem 4** *Algorithm 1 is $O(n|V|)$ for a directed plane graph $\vec{G}(\mathcal{F}) = (V, E, F)$ where $\mathcal{F}$ is a monotone FISC with $n$ curves.*

**Algorithm 1** Let $\mathcal{F}$ be a montone FISC. Given $\vec{G}(\mathcal{F})$ and weight function $\omega$ for the faces, this algorithm produces a radial drawing of $\vec{G}(\mathcal{F})$ where the faces have areas proportional to $\omega$.

1: {Compute the directed dual of $\vec{G}(\mathcal{F})$.}
2: $\vec{D}(\mathcal{F}) \leftarrow$ directed plane dual of $\vec{G}(\mathcal{F})$
3: $v(s) \leftarrow$ the unique source vertex of $\vec{D}(\mathcal{F})$
4: $v(t) \leftarrow$ the unique sink vertex of $\vec{D}(\mathcal{F})$
5: {Compute $h(f)$ and $t(f)$ for each face $f$ of $\vec{G}(\mathcal{F})$.}
6: **for all** faces $f \in \vec{G}(\mathcal{F})$ **do**
7: $\quad h(f), t(f) \leftarrow$ vertex from which the two paths of $f$ converge,diverge
8: **end for**
9: {Make $\vec{G}(\mathcal{F})$ acyclic by removing edges between common face and empty face.}
10: $P \leftarrow$ any path from $v(s)$ to $v(t)$
11: **for all** edges $e \in P$ **do**
12: $\quad$ remove the respective dual edge $e'$ from $\vec{G}(\mathcal{F})$
13: **end for**
14: {Perform a layered topological sort of $\vec{G}(\mathcal{F})$.}
15: $k \leftarrow 0$
16: $L_1 \leftarrow$ source vertices of $\vec{G}(\mathcal{F})$
17: **while** $|L_{k+1}| > 0$ **do**
18: $\quad k \leftarrow k + 1$
19: $\quad$ **for all** vertices $v \in L_k$ **do**
20: $\quad\quad$ remove $v$ and its incident edges from $\vec{G}(\mathcal{F})$
21: $\quad$ **end for**
22: $\quad L_{k+1} \leftarrow$ source vertices of $\vec{G}(\mathcal{F})$
23: **end while**
24: {Draw the faces outward from the common face.}
25: $x \leftarrow$ an arbitrary point in the plane
26: $p_1, p_2, \ldots, p_{2k} \leftarrow$ the clockwise points of a $2k$-gon with area $\omega(s)$ that contains $x$
27: Draw polygon $p_1, p_2, \ldots, p_{2k}$
28: **for all** bfs-order vertices $v(f) \in \vec{D}(\mathcal{F}) \backslash \{v(s), v(t)\}$ **do**
29: $\quad i, j \leftarrow$ indices where $t(f) \in L_{\lfloor (i+1)/2 \rfloor}, h(f) \in L_{\lfloor (j+1)/2 \rfloor}$
30: $\quad$ {NOTE: points are indexed, modulo $2k$, beginning at $p_1$.}
31: $\quad$ Draw polygon $p_i, p'_{i+1}, \ldots, p'_{j-1}, p_j, p_{j-1}, \ldots, p_{i+1}$ where these are the clockwise points of a polygon with area $\omega(f)$ that is contained within the sector defined by $x, p_i, p_j$ and does not overlap any existing polygons.
32: $\quad p_{i+1}, p_{i+2}, \ldots, p_{j-1} \leftarrow p'_{i+1}, p'_{i+2}, \ldots, p'_{j-1}$
33: **end for**

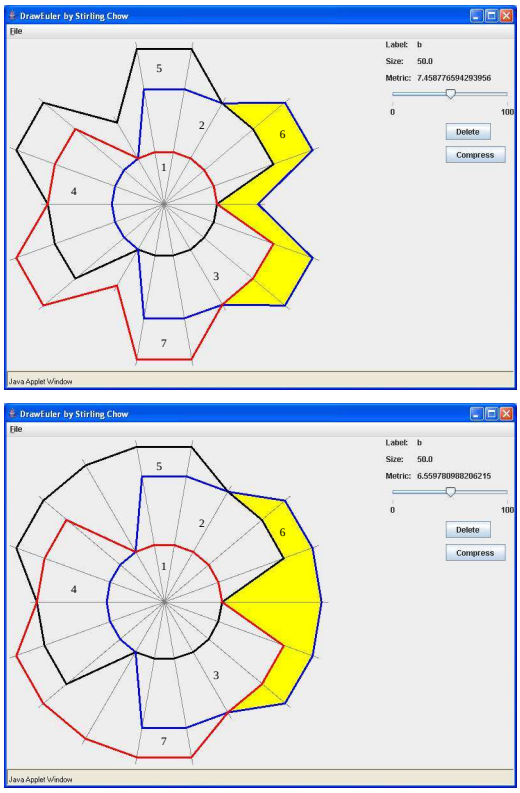


Figure 2: The result of Alg. 1 on the FISC from Fig. 1(a) using two different polygon growth formulae.

**Proof.** We assume $\vec{G}(\mathcal{F})$ is represented using the Winged-Edge data structure [2] which provides $O(1)$ operations for inserting and removing edges, getting an arbitrary edge adjacent to a vertex or face, getting the endpoints of an edge, getting the left and right faces of an edge, and getting the predecessor and successor edges about either of an edge's adjacent faces.

Lines 2–4. The directed dual is created by traversing the edges of each face of $\vec{G}(\mathcal{F})$ and inserting a respective dual edge about a new vertex in $\vec{D}(\mathcal{F})$; since each edge is visited twice, this is an $O(|F|+|E|)$ construction. $v(s)$ and $v(t)$ can be kept track of during the construction.

Lines 6–8. Similarly, the edges of each face in $\vec{G}(\mathcal{F})$ are traversed and the vertices where two consecutive edges converge/diverge is recorded; this is an $O(|F| + |E|)$ operation.

Line 10. In constructing $P$, an edge directed out of the current vertex must be found by searching all the incident edges. Since $D$ is acyclic, each edge is visited at most once, so this is an $O(|E|)$ construction.

Lines 11–13. Since $P$ is a directed path from the common face to the empty face, it visits at most $n-1$ edges. Since edge removal is $O(1)$, the complexity of these steps is $O(n)$.

Lines 15–23. A topological sort is $O(|V| + |E|)$.

Lines 25–33. We assume that computing a polygon

involves solving a piecewise function, the size of which depends on the number of spanned rays. Since $\mathcal{F}$ is ray-monotone, each ray intersects at most $n$ polygons, and since there are at most $2|V|$ rays, the number of equations that must be solved is at most $2|V|n$. If each equation is solved in constant time (which is the case for the polygon growth methods described), the complexity of these steps is $O(n|V|)$.

The overall complexity is $O(|F|+|E|)+O(n)+O(|V|+|E|)+O(n|V|)$. Euler's Formula linearly relates $|V|, |E|$, and $|F|$, so this simplifies to $O(n|V|)$ for $n \geq 2$. □

## 5 Remarks

Another way to draw area-proportional FISCs is to use existing cartogram algorithms. In this scenario, a planar graph drawing algorithm (e.g., Tutte or spring-embedding) would be used to produce a *non-area-proportional* drawing of $\vec{G}(\mathcal{F})$ which would subsequently be used as input to a cartogram algorithm. Cartogram algorithms are based on an iterative optimization strategy where tradeoffs are made between achieving the required area-proportionality and maintaining similarity to the input drawing (e.g., global outline or local angle preservation). Because of this optimization, the algorithms are compute-intensive (sometimes taking hours, although recent work has produced faster methods [9]), and often require numerical methods for solving PDEs. As such, they are usually unsuitable for realtime exploration (e.g., dynamically adjusting populations).

Since monotone FISCs are a special case of cartogram input, our algorithm can take advantage of their properties to implement very fast computation of an area-proportional drawing. In fact, we have implemented Alg. 1 as a Java applet that allows the region populations, the ray distribution, and the polygon growth formulae to be changed in realtime (the screenshots come from this applet). An avenue for further research is to use our algorithm's drawings as input to a cartogram application, and define optimization criteria based on desirable aesthetic qualities. It may be the case that since area-proportionality has already been achieved, the cartogram algorithm may converge more rapidly.

Finally, our interest in drawing area-proportional FISCs was motivated by the problem of drawing area-proportional Euler diagrams. In this scenario, no FISC is initially provided; instead, the input to the problem is a set system that specifies which regions of a FISC are desired (e.g., {{a},{b},{ab},{abc}}). Algorithm 1 can be used to solve this problem by first generating a monotone Venn diagram FISC (using Anthony Edwards' construction [7]), and then drawing the FISC so that unwanted regions are given zero weight. In this way, an unwanted face's free path collapses on its fixed path to make the region disappear (see Fig. 3). Various postprocessing can be applied to remove extraneous overlapping edges as detailed in [5].


Figure 3: The result of deleting region b from Fig. 2(b) and manipulating the rays and populations.

Readers are encouraged to try our implementation of the algorithm which is available at the authors' website: `http://theory.cs.uvic.ca/venn/DrawEuler/`.

## References


[1] P.H. Artes and B.C. Chauhan. Longitudinal changes in the visual field and optic disc in glaucoma. Prog. in Retinal and Eye Research, Articles in Press, Cor. Proof.

[2] B. Baumgart. Geometric Modelling for Computer Vision. *Ph.D. Thesis*, Stanford University, 1974.

[3] B. Bultena, B. Grünbaum, and F. Ruskey. Convex drawings of intersecting families of simple closed curves. In *Proc. 11th Cdn. Conf. on Comp. Geom.*, pages 18–21, 1999.

[4] S. Chow and F. Ruskey. Drawing area-proportional Venn and Euler diagrams. In *Proc. of Graph Drawing 2003*, Spring-Verlag LNCS 2912, pages 466–477, 2004.

[5] S. Chow and F. Ruskey. Towards a general solution to drawing area-proportional Euler diagrams. In *Proc. of Euler Diagrams 2004*, Electronic Notes in Theoretical Computer Science, 134, pages 3–18, 2005.

[6] H. Edelsbrunner and E. Waupotitsch. A combinatorial approach to cartograms. In *Comp. Geom. Theory and Applications*, 7, pages 343–360, 1997.

[7] A.W.F. Edwards, *Cogwheels of the Mind: The Story of Venn Diagrams*, The Johns Hopkins University Press, 2004.

[8] L. Euler, *Lettres á une Princesse d'Allemagne sur Divers Sujets de Physique et de Philosophie*, Imperial Academy of Sciences, Petersburg, 1768–1772.

[9] D. Keim, S. North, and C. Panse. Cartodraw: A fast algorithm for generating continuous cartograms. In *IEEE Trans. on Visualization and Computer Graphics*, 10, pages 95–110, 2004.

[10] W. Tobler. Thirty Five Years of Computer Cartograms. In *Annals of the Assoc. of Amer. Geog.*, 94 (1), pages 58–71, 2004.