Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 19, no. 1, pp. 393–412 (2015)
DOI: 10.7155/jgaa.00366

Drawing Graphs with Few Arcs

André Schulz

LG Theoretische Informatik, FernUniversität in Hagen, Germany

Abstract

Let $G = (V, E)$ be a planar graph. An arrangement of circular arcs is called a *composite arc-drawing* of G, if its 1-skeleton is isomorphic to G. Similarly, a *composite segment-drawing* is described by an arrangement of straight-line segments. We ask for the smallest possible ground set of arcs/segments for a composite arc/segment-drawing. We present algorithms for constructing composite arc-drawings with a small ground set for trees, series-parallel graphs, planar 3-trees and general planar graphs. In the case where G is a tree, we also introduce an algorithm that realizes the vertices of the composite drawing on a $O(n^{1.81}) \times n$ grid. For each of the graph classes we provide a lower bound for the maximal size of the arrangement's ground set.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
December 2013	July 2015	August 2015	August 2015	August 2015
		Published: August 2015		
	Article type: Regular Paper		Communicated by: A. Symvonis	

The research was performed in cooperation with Eurogiga project GraDR - Graph Drawings and Representations. It was partially funded by the German Research Foundation DFG under grant SCHU 2458/4-1. A preliminary version has appeared at the *39th InternationalWorkshop on Theoretic Concepts in Computer Science* [14].

E-mail address: andre.schulz@fernuni-hagen.de (André Schulz)

1 Introduction

There exists a large number of design criteria for *good drawings* of planar graphs such as small area, good vertex and angular resolution, or a small number of edge crossings. All these measures assure that vertices and edges in a drawing are distinguishable for the observer. In this paper we propose a novel criterion for aesthetic and readable graph drawings. Our goal is to generate drawings that are easy to *perceive* by the viewer. When reading a drawing the human mind decomposes the received picture into geometric entities such as lines, segments, arcs, disks, circles, and so on. By interpreting the relationship between these entities an understanding of the drawing is obtained. We refer to the number of entities used in the drawing as its *visual complexity*.

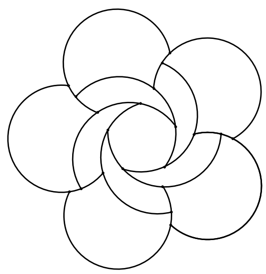
Straight edges and the absence of crossings are desirable features for a drawing. A straight edge would be considered as one single entity, whereas, for example, a polygonal chain might be considered as a combination of several geometric entities. Something similar is true for edge crossings. If two edges cross, they introduce a new *perceptional feature* in the drawing – the crossing point. Therefore, a noncrossing straight-line drawing is formed by a combination of $|V| + |E|$ geometric entities. In this paper we want to reduce the number of geometric entities even further. To make this possible, we group edges, such that they form a new entity. For example, if we are able to draw a path of the graph as a single straight-line segment in the drawing (with vertices in its interior), the visual complexity of the drawing is reduced. More formally, we define:

Definition 1 (Composite drawing) *Let $\mathcal{A}$ be an arrangement of simple geometric bounded 1d objects in the plane. The objects might be subdivided by placing additional vertices on them. Let G be the 1-skeleton of the subdivided arrangement. The arrangement $\mathcal{A}$ is called a* composite drawing *of G. If $\mathcal{A}$ contains only line segments it is called a* composite segment-drawing*, if $\mathcal{A}$ contains also circular arcs it is called a* composite arc-drawing*. The number of arcs/segments of $\mathcal{A}$ refers to the cardinality of the ground set of $\mathcal{A}$.*

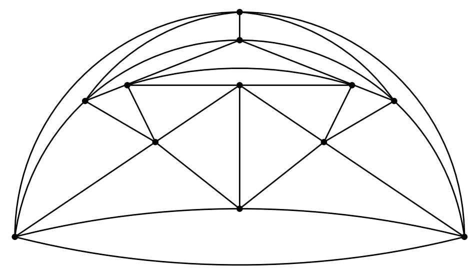
Fig. 1 shows examples of composite arc-drawings.

Our motivation for the perception based approach stems partially from the work of the artist Mark Lombardi. Lombardi's visual art was focused on graph drawings of social networks within the political and financial sector [10]. The drawings of Lombardi have a unique style. Maybe the most characteristic feature is the use of circular arcs to represent consecutive edges. These circular-arc paths kept the visual complexity of the drawings low. By aligning edges Lombardi enhanced his drawings with additional information. For example, these alignments were used to decode temporal or sequential dependencies of events represented by the vertices.

In this work, we focus on the combinatorial aspects of drawings with low visual complexity. As simple geometric objects for composite drawings we consider (straight-line) segments and circular arcs. Using straight-line segments



(a)



(b)

Figure 1: A drawing with low visual complexity of the graph of the dodecahedron (a). The drawing uses 10 circular arcs, which is the best possible. A drawing of the icosahedron graph that has not the lowest possible visual complexity (b).

graph class	upper bounds segments [6]	upper bounds arcs	lower bounds segments	
trees	$\lceil \lvert E\rvert/2\rceil$	$\lceil \lvert E\rvert/2\rceil$	$\lceil \lvert E\rvert/2\rceil$	[6]
— on $O(n^{1.81}) \times n$ grid	–	$\lceil 3\lvert E\rvert/4\rceil$	$\lceil \lvert E\rvert/2\rceil$	Thm. 1
series-parallel	$3\lvert E\rvert/4+1$	$\lvert E\rvert/2+1$	$\lvert E\rvert/4$	Thm. 2
planar 3-trees	$2\lvert E\rvert/3+4$	$11\lvert E\rvert/18+3$	$\lvert E\rvert/6$	Thm. 3
planar 3-connected	$5\lvert E\rvert/6+2$	$2\lvert E\rvert/3$	$\lvert E\rvert/6$	Thm. 4

Table 1: Combinatorial results obtained in the paper. The lower bounds in the third and fifth row are presented as slightly simplified expressions.

is the most natural way for drawing edges, but also circular arcs have been proposed as "edge shapes" before [1, 5]. In our understanding, a line segment is a degenerated circular arc, and by a suitable Möbius transformation these segments can be converted to proper circular arcs. We present bounds on the maximal number of arcs/segments necessary in a composite drawing. Our approach cannot handle edge crossings, since every crossing defines a subdivision of geometric objects and hence introduces an additional vertex in the composite drawing. Therefore we only study (noncrossing) drawings of planar graphs, such as trees, series-parallel graphs, and planar 3-trees. Moreover, all graphs that we consider are simple, which means that we forbid parallel edges and self-loops. The results of this paper are listed in Tab. 1. All lower bounds presented in this paper are due to the following simple lemma [6].

Lemma 1 *Let G be a graph with N vertices of odd degree. Every composite arc-drawing or segment-drawing of G requires at least $N/2$ arcs.*

Proof. In every odd degree vertex at least one arc/segment has an endpoint. Hence we have at least N endpoints of arcs. Since the number of odd vertices is an even number the statement of the lemma follows. □

Related Work. Dujmović et al. [6] studied the complexity of composite segment-drawings. They presented their results in a slightly different form, namely, the bounds on the number of segments are expressed in terms of $|V|$, instead in terms of $|E|$. We are however convinced that a bound in terms of $|E|$ gives a more universal expression since a graph with fewer edges tends to require fewer segments or arcs. The results of Dujmović et al. are presented in Tab. 1. Our results for composite arc-drawings are an improvement over the (straight-line segment) bounds of Dujmović et al.. None of the drawings of Dujmović et al. fulfilled additional aesthetic quality criteria. In fact, they stated the problem of designing algorithms with small area as an open problem. From this perspective, Theorem 1 gives the first algorithm that constructs composite drawings on a small polynomial grid.

User studies comparing straight-line drawings with circular-arc drawing have been conducted only recently [12, 16]. Both studies showed that certain tasks are easier to carry out by the observer, when straight edges are used. On the other hand, users preferred the aesthetics of circular arc drawings over straight-line drawings in one of the studies [12]. Note that these studies have not considered drawings with low visual complexity, but only drawings with circular arcs. The hypothesis that drawings with low visual complexity are indeed easier to perceive still needs to be checked empirically, which is work in progress.

Recently, so-called smooth orthogonal layouts have been studied [2, 3]. Instead of polyline edges, which are typically used in orthogonal layouts, chains of smooth circular arcs are proposed as edge shapes. The goal is to optimize the maximal edge complexity (number of circular arc pieces per edge) and not the visual complexity of the whole drawing.

2 Composite drawings of trees

Let $T = (V, E)$ be a tree that we want to realize as a composite segment-drawing. Drawings with $\lceil |E|/2 \rceil$ segments can be constructed by a greedy algorithm [6], which is optimal.

2.1 Grid drawings of trees with few arcs

In this subsection we show how to draw an unordered tree as a composite arc-drawing with few arcs and the additional constraint that all vertices lie on the $\mathbf{Z}^2$ grid. Our objective is to obtain a drawing that uses few arcs but also requires a small grid. Note that the greedy algorithm in [6] yields an embedding on a grid exponential in $|V|$.

To obtain a drawing on a small grid we do not aim at drawings with the *lowest* visual complexity. We believe that both grid size, and visual complexity cannot be optimized at the same time. As an easy example, the reader might consider the realization of a simple cycle. Obviously this graph can be drawn

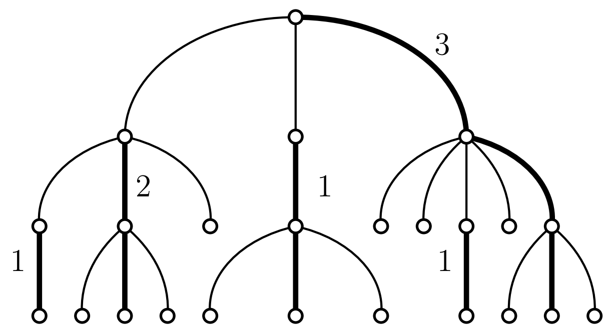


Figure 2: A tree with its heavy paths (indicated by the thick edges). The numbers refer to the depth of the heavy paths.

with only one circle. However realizing a circle such that it contains many grid points is a highly nontrivial task. To our knowledge the best method uses a grid of size $O(5^{n/4})$ [13].

2.1.1 Heavy path decomposition

The drawing algorithm is based on a decomposition scheme for trees, called the *heavy path decomposition* [15], which works as follows. We root the tree $T = (V, E)$ at some vertex r. Let u be a node of T, then T_u denotes the subtree rooted at u, and $N(u)$ denotes the size of this subtree. For every non-leaf u we select a child v, for which $N(v)$ is maximal (with respect to the size of the subtrees of the other children). The edge (u, v) is called a *heavy edge* and all edges that are not heavy are called *light edges.* A maximal connected component of heavy edges is called a *heavy path.* The tree T decomposes into heavy paths and light edges. Note that every path in T to the root visits at most $\lceil \log |V| \rceil$ light edges.

For the drawing algorithm it is convenient to introduce the following definitions. We call the node on a heavy path that is closest to r its *top node.* The subtree rooted at the top node of a heavy path its called the *heavy path subtree* of this path. The light edge that links the top node of a heavy path P with its parent in T is called *light parent edge* of P. The *depth* of a heavy path P is defined as follows: If P is not incident to light parent edges of other heavy paths, it has depth one. Otherwise we obtain the depth of P by adding one to the maximal depth of a heavy path linked to P via its light parent edge. Note that the subtrees of heavy paths of a fixed depth are all disjoint. Figure 2 shows an example of a heavy path decomposition with annotated depth of the heavy paths.

2.1.2 Algorithm outline

The drawing algorithm works (high-level) as follows. We draw all subtrees of heavy paths with increasing order of their depth. Furthermore, we associate every subtree of a heavy path with an axis-aligned rectangle called its *safe box*. As an invariant the drawing of a subtree is exclusively contained inside its safe box, and the root of the subtree is placed on the top edge of its safe box, but not on its corners. For convenience we require that every safe box has width at least 3, in particular every leaf is placed inside a 3×1 safe box. When going from a depth k to a depth $k+1$ subtree we arrange the drawings of the subtrees whose heavy paths have smaller depth to a new drawing (details will be given later). The algorithm terminates when the heavy path subtree with the largest depth has been drawn.

Let us explain the recursive step of the algorithm, that is, how to build the subtrees of the heavy paths (see Fig. 3(a) for an illustration). The heavy path is drawn as a single vertical segment. The only exception might be its edge (u, v) incident to the leaf v. Note that every subtree incident to u has to be a leaf as well, otherwise (u, v) would not be a heavy edge. Hence all k children of u are leaves, a node with this property is called a *k-fork* in the following. The children of u are placed on the line $y = 0$ and u is placed on $(0, 1)$. In the case that k is even, we place the children of u symmetrically around the y-axis such that they have x-coordinates $-k/2, -k/2+1, \ldots, -1, 1, \ldots k/2-1, k/2$. Two vertices are joined by an arc through u when they have the same absolute x-coordinate (see Fig. 3(a)). In case that k is odd we place the light edges as in the even case and realize the heavy edge (u, v) by extending the vertical segment that contains the remaining heavy path (see Fig. 3(b)).

Assume now that u is not a fork. All safe boxes of subtrees incident to u will be drawn, such that their roots lie on the same horizontal line which is one unit below u. Moreover, they will be distributed, such that two of them are connected by a single arc running through u. Note that if we have an odd number of light edges for u, one of the safe boxes does not have a sibling to pair with. In this case we draw the arc as if there would be a sibling (leaf) but we draw only the half of the arc that connects to v. The location of the safe boxes incident to u needs vertical space, which is determined by the safe box with the largest height. The smallest horizontal strip containing all safe boxes incident to u is called a *row*. The tree is constructed such that all of its rows are separated vertically by one unit. The child w following u on the heavy path is placed at the bottom boundary of the row directly below u.

2.1.3 Box displacement

We now discuss how to arrange the safe boxes within each row. Let u be a node on the heavy path P (not a leaf or fork) and let $v_1, v_2, \ldots, v_k$ be the k children of u not on P. By recursion, the subtrees rooted at the v_is have already been drawn, so we have for every v_i a safe box B_i with width w_i and height h_i. Recall that v_i is placed on the top edge of B_i. We will arrange all safe boxes B_i such that their top edges lie on a common horizontal line, the node v_i has x-coordinate x_i, and the node u is placed one unit above at $x = 0$. To draw multiple light edges with a single arc, we pair two children, say v_i and v_j, and connect both by an arc running through u. This implies that $x_i = -x_j$ for every

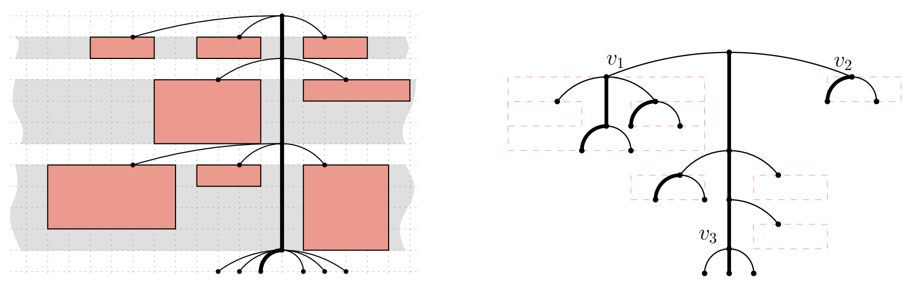


Figure 3: (a) A drawing of a heavy path's subtree. Rows are drawn shaded and the heavy path is drawn thick. (b) An example of a composite arc-drawing of a tree. The safe boxes used in the algorithm are indicated by dashed rectangles. Vertex v_1 is a hep, v_2 is a 2-fork and, v_3 is a 3-fork.

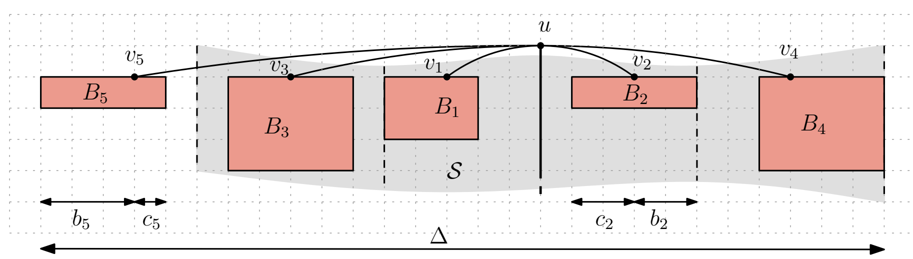


Figure 4: A snapshot during the execution of the greedy strategy for the box displacement. The safe boxes up to B_4 have already been placed. When placing the last box B_5 we avoid the restricted strip $\mathcal{S}$. The boundaries of the strip $\mathcal{S}$ after round 1 and 2 are drawn as dashed lines.

such pair of vertices.

We determine the location of the safe boxes by a greedy strategy (see Fig. 4). Let ℓ_i be the distance from v_i to the top left corner of B_i, and similarly, let r_i be the distance from v_i to the top right corner of B_i. We first orient all boxes such that $\ell_i \geq r_i$ (it is valid to reflect the whole safe box including the drawing). Then we sort the boxes by ℓ_i in increasing order, and finally we flip all boxes with an even index vertically, such that $r_i \geq \ell_i$.

Assume for now that k is an even number. We place the safe boxes in *rounds*. In round t we place the safe boxes B_{2t-1} and B_{2t}, $t = 1, 2, \ldots k/2$, and connect them by an arc passing through u. For convenience we introduce the following notation: If a box B_i is placed left of the heavy edge, then $c_i := r_i$ and $b_i := \ell_i$, otherwise $c_i := \ell_i$ and $b_i := r_i$. This implies that $b_i \geq c_i$ for $i \leq k$.

In the first round we place B_1 and B_2. Without loss of generality we assume that $c_1 \leq c_2$ (otherwise the strategy is symmetric). We place B_2 as close as possible to the line $x = 0$. Since no safe boxes have been placed before, we only have to avoid the heavy edge emanating from u. Hence, the safe box is placed such that $x_2 = c_2 + 1$. Next, we place B_1. The location of x_1 is already determined since we have fixed x_2. Note that B_1 is guaranteed to be to the left of the heavy edge emanating from u, since $c_2 \geq c_1$. Let $\mathcal{S}$ be vertical strip with smallest width centered at the y-axis that contains the safe boxes placed so far. In the following rounds we place the remaining safe boxes such that they are separated from $\mathcal{S}$ by one unit and update $\mathcal{S}$ after every round.

In case k is odd, only one safe box needs to be placed in the final round. We draw the final safe box on the left side, such that it is separated from $\mathcal{S}$ by one unit. When all safe boxes have been arranged we determine the *width of the displacement* Δ, that is the distance between the most extreme top corners. The only exception is when $k = 1$; in this case Δ equals the width of the only safe box plus 2 as shown in Fig. 5.

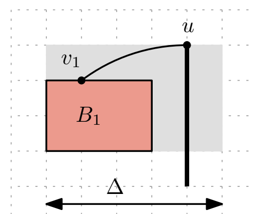


Figure 5: The special case where u has only one child connected by a light edge.

Lemma 2 *Assume we carried out the box displacement for the safe boxes incident to some u on P by the greedy strategy as explained above. We have*

$$\Delta \leq 7/4 \sum_{i=1}^{k} w_i.$$

Proof. Let $X_c = \sum_i^k c_i$ and $X_b = \sum_i^k b_i$. We have, $X_c + X_b = \sum_i^k w_i$. Due to our preprocessing step we know that (i) $b_i \geq c_i$, and (ii) the safe boxes are sorted such that $b_{i+1} \geq b_i$ for all $i < k$. As a consequence of (i) we have that $X_c \leq X_b$.

Let us first check the case $k = 1$. To maintain the invariant, that the root does not lie on a corner of the safe box, we have to extend the width Δ (see Fig. 5). We obtain $\Delta = 2 + w_1$, which is smaller than $7/4w_1$ for $w_1 \geq 3$. Since we placed all leaves inside 3×1 boxes the statement of the lemma holds for this case.

Assume now that k is even. In every nonterminal round i the width of the strip $\mathcal{S}$ increases by $2(1+\max\{b_{2i-1}, b_{2i}\}+\max\{c_{2i-1}, c_{2i}\})$. In the last round we determine the width of the displacement Δ by extending the width of the current strip $\mathcal{S}$ by $2+2\max\{c_{k-1}, c_k\}+b_{k-1}+b_k$. We know that $\max\{b_{2i-1}, b_{2i}\} = b_{2i}$.

Summing up all rounds yields

$$\begin{aligned}
\Delta &= 2\sum_{i=1}^{k/2-1}(1+b_{2i}+\max\{c_{2i-1},c_{2i}\})+2+2\max\{c_{k-1},c_k\}+b_{k-1}+b_k\\
&\le 2\sum_{i=1}^{k/2-1}(b_{2i}+c_{2i-1}+c_{2i})+2c_{k-1}+2c_k+b_{k-1}+b_k\\
&\le 2X_c+X_b+\sum_{i=1}^{k/2-1}(b_{2i}-b_{2i-1})\\
&< 2X_c+X_b+b_{k-2}\\
&\le 5/3\sum_{i=1}^{k}w_i.
\end{aligned}$$

The second to last inequality follows by the monotonicity of $(b_1, b_2, \ldots, b_k)$. For the last transition we have used the facts that $X_b + X_c = \sum_i w_i$ and due to the ordering $b_{k-2} \le X_b/3$.

Finally, if k is odd and $k \neq 1$ we have to change the above estimation only marginally. For convenience, we assume that k is still an even number, but we place only $k-1$ safe boxes. We obtain

$$\begin{aligned}
\Delta &= 2\sum_{i=1}^{k/2-1}(1+b_{2i}+\max\{c_{2i-1},c_{2i}\})+1+c_{k-1}+b_{k-1}\\
&\le 2\sum_{i=1}^{k/2-1}(b_{2i}+c_{2i-1}+c_{2i})+c_{k-1}+b_{k-1}\\
&\le 2X_c+X_b+\sum_{i=1}^{k/2-1}(b_{2i}-b_{2i-1})\\
&< 2X_c+X_b+b_{k-2}\\
&\le 7/4\sum_{i=1}^{k}w_i.
\end{aligned}$$

Since there are only $k-1$ safe boxes, we have to use the weaker bound $b_{k-2} \le X_b/2$ for the last transition.

□

Let t be the top node of a heavy path P. After carrying out the box displacements for all rows we can define the safe box for the subtree of P. Its width is determined by the row with the maximal displacement width. By construction, t lies on the top edge, but not on a corner of the new safe box. Fig. 4 shows an example of the greedy strategy.

Lemma 3 *By inductively laying out the safe boxes with the greedy strategy explained above, the heavy path decomposition yields a drawing where every vertex is placed on a $O(n^{1.81}) \times n$ grid.*

Proof. In every inductive step we construct a drawing of a subtree and its safe box out of smaller safe boxes. Assume we have k such safe boxes $B_1, \dots, B_k$. Due to Lemma 2 the width of the new safe box is at most $7/4 \sum_{i=1}^{k} w_i$, since it might happen that all safe boxes are placed in one row. On the other hand, at least one box is placed in every row, and these rows are vertically separated by one unit. This shows that the height of the new safe box is at most $m + \sum_{i=1}^{k} h_i$, for m being the number of rows plus one.

The claim of the lemma follows by induction. We first discuss the height. When a small box contains only a single vertex, its height is one. When combining the small boxes to a new subtree, we obtain as new height $m + \sum_{i=1}^{k} h_i$. This new subtree, however, has at least the vertices contained in the smaller safe boxes and the m vertices on its heavy path. Hence the height of its safe box is at most the number of its vertices.

For the width we notice that due to the heavy path decomposition the recursion depth is at most $\lceil \log n \rceil$. By induction a subtree of a heavy path with depth d and n' vertices is contained inside a safe box of width at most $3 \cdot (7/4)^d \cdot n'$. Hence the whole tree is contained in a box of width $3n \cdot (7/4)^{\lceil \log n \rceil}$ which is upper bounded by $O(n^{1.81})$. □

2.1.4 Analysis

A node is called a *heavy even-prefork* (short *hep*), if its heavy edge child is a k-fork, with k even. Fig. 3(b) illustrates the definition. A charging scheme for the "saved edges" in forks and heps leads to the following lemma.

Lemma 4 *Let $T = (V, E)$ be a tree drawn as a composite arc-drawing with the algorithm based on the heavy path decomposition. Then the drawing uses at most $\lceil 3|E|/4 \rceil$ arcs.*

Proof. For technical reasons we introduce a virtual edge, from the root of the tree T to an imaginary father, such that every node has a father. This only increases the number of edges by 1. Let F_k be the set of k-forks in T, and let E_k denote the $(k+1)$ edges incident to a k-fork. We define $E_{\text{odd}} := \bigcup_{k \text{ odd}} E_k$ and $E_{\text{even}} := E \setminus E_{\text{odd}}$. The edges for all forks are disjoint and hence we have $|E| = |E_{\text{odd}}| + |E_{\text{even}}|$. Finally, we denote by h the number of heps. For every subtree S we define a potential $\Phi(S)$ based on the composite drawing restricted to S. We set

$$\Phi(S) := 2(\#\text{arcs in } S) - (\#\text{edges in } S).$$

If the root s of S is a k-fork, then S is a star with center s, and we have

$$\Phi(S) = \begin{cases} 0 & \text{if } k \text{ is even} \\ 1 & \text{if } k \text{ is odd} \end{cases}.$$

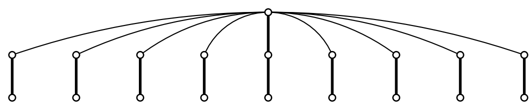


Figure 6: An example where the analysis of Lemma 4 is tight.

If S is a single node (e.g., a leaf of T) we have $\Phi(S) = 0$. Assume now that s is neither a fork nor a leaf. Let $S_1, S_2, \ldots, S_t$ be the subtrees rooted at its children. There are $t-1$ light edges to children of s, which are drawn using $\lfloor t/2 \rfloor$ arcs. Let e be the heavy edge that joins s with some S_i. If s is not a hep, then e can be drawn for free, because the vertical segment that represents the heavy path (except maybe for the last edge) was already partially drawn in S_i. Thus, e can be realized by extending this "heavy path part". On the other hand, if s is a hep, we have to draw e as a new arc. Let $\chi(S)$ be 1 if S is a hep, and 0 otherwise. We obtain

$$\Phi(S) \leq \sum_{i=1}^{t} \Phi(S_i) + 2\chi(S).$$

This implies

$$\Phi(T) \leq \sum_{k\colon k \text{ odd}} |F_k| + 2h. \tag{1}$$

We now assign edges to a hep as follows: for every hep we select the two heavy edges in its subtree, one of the light edges of the fork, and the parent edge, which gives 4 edges in total. Note that the edges associated to a hep are neither assigned to another hep nor contained in a k-fork with odd k. As a consequence we have $h \leq |E_{\text{even}}|/4$.

We apply our edge charging scheme to Eq. (1) and observe that since $|E_k|/(k+1) = |F_k|$ it holds that $\sum_{k\colon k \text{ odd}} |F_k| \leq |E_{\text{odd}}|/2$. This gives $\Phi(T) \leq |E_{\text{odd}}|/2 + |E_{\text{even}}|/2 = |E|/2$, and therefore no more than $3|E|/4$ arcs are used in the composite drawing. Note that we had introduced an additional edge to T. If we subtract this edge, we obtain that there are at most $\lceil 3|E|/4 \rceil$ arcs in the composite drawing for T. □ Note that the bound stated in Lemma 4 is tight for trees that are stars with subdivided edges as shown in Fig. 6. Combining Lemma 2, 3, 4 yields Theorem 1.

Theorem 1 *The algorithm for realizing a tree $G = (V, E)$ as composite arc-drawings uses at most $\lceil 3|E|/4 \rceil$ arcs. The computed drawing realizes all vertices on a $O(n^{1.81}) \times n$ grid, for $n = |V|$.*

3 Composite drawings of serial-parallel graphs and planar 3-trees

In this section we study composite arc-drawings of series-parallel graphs and planar 3-trees. We start with series-parallel graphs.

A graph $G = (V, E)$ is a series-parallel graph if it can be built from a sequence of serial or parallel combination steps [7]. We do not give the details of this standard definition but use the following alternative definition instead. A graph is a *series-parallel* graph if it can be decomposed into a sequence of paths $E_1, E_2, \ldots E_k$, such that (1) the endpoints for every path not E_1 lie both on some path E_j with smaller index (the path between the two endpoints on E_j is called a *nested interval*), (2) no interior point of a path is contained in a path with smaller index, and (3) all nested intervals are either disjoint or contain each other [7]. Such a decomposition is called a *nested open ear-decomposition.*

Theorem 2 *Let $G = (V, E)$ be a series-parallel graph. Based on a nested open ear-decomposition we can obtain a composite arc-drawing with at most $(|E| + 1)/2$ arcs. For every n there is a series-parallel graph $G = (V, E)$ with more than n vertices, whose composite segment-drawings need at least $(|E| + 2)/4$ segments.*

Proof. Let $E_1, E_2, \ldots E_k$ be a nested open ear-decomposition of G. As noted by Miller and Ramachandran [11], we have $k = |E| - |V| + 2$. We first draw E_1 as single segment and then draw the other paths in increasing order as circular arcs. Suppose that we draw the path E_i with endpoints on E_j with $j < i$. By construction, E_j has been realized as a circular arc. Let E'_j be a copy of the part of E_j that lies between the two endpoints of E_i. We can draw E_i "on top" of E_j by slightly increasing the radius of E'_j. The angle of the tangents of E_i and E_j at their intersection can be made arbitrarily small. Hence, when executing the drawing process, we can assume that every circular arc has small curvature. Since all paths are "nested" we can finish the drawing without introducing an edge crossing. Fig. 7(a) shows a drawing obtained by the algorithm.

Any (simple) series-parallel graph has $|E| \le 2|V| - 3$ edges. The drawing uses exactly k arcs. This gives

$$\#\text{ arcs} = k = |E| - |V| + 2 \le |E| - \frac{|E| + 3}{2} + 2 = \frac{|E| + 1}{2}.$$

For the lower bound consider a graph as depicted in Fig. 7(b). More generally, we consider for some odd $n > 3$ the nested open ear-decomposition given by $E_1 = (v_1, \ldots, v_n)$ and $E_i = (v_1, v_{i+1})$, for $2 \le i \le n - 2$. In such an n-vertex graph only v_2 has even degree and thus we have $n - 1 = |E|/2 + 1$ odd degree vertices. Lemma 1 implies that such a graph needs at least $|E|/4 + 1/2$ arcs in any composite arc-drawing. □

The next class of graphs we consider are the planar 3-trees. A planar 3-tree is a triangulation that can be defined recursively as follows: Suppose $G =$

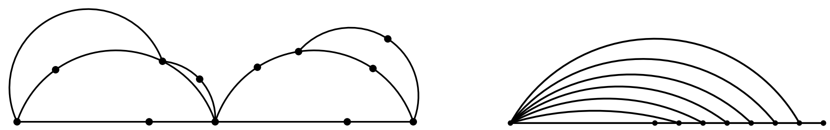


Figure 7: (a) A composite arc-drawing of a series-parallel graph, obtained by the method explained in the proof of Theorem 2. (b) A series-parallel graph with only one even-degree vertex.

$(\{v_1, \ldots, v_n\}, E)$ is a triangulation, we can pick one of its faces, say it is spanned by the vertices v_i, v_j, v_k and add a new vertex u inside this face together with the three edges connecting v_i, v_j, v_k with u. By this we remove one face and introduce 3 new faces. This operation is called a *stacking operation*. Any graph that can be generated from a triangle by a sequence of stacking operations is called a *planar 3-tree*. We say that a planar 3-tree is a *k-fan* if it has $k+3$ vertices and it contains the triangle v_1, v_2, v_3 and for every $4 \leq i \leq k+3$ the edges $(v_i, v_1), (v_i, v_2)$, and (v_i, v_{i-1}).

To develop an algorithm for a composite arc-drawing we first introduce a crucial lemma. For the lemma we need the following definitions. A triangle is called *spherical* if its edges are circular arcs that do not intersect. If furthermore every triangle corner is incident to an angle (measured between tangents) larger than zero and less than π, we call the spherical triangle *nonreflex*. We say a vertex v inside a spherical triangle S is *spherically visible* from a point c, if there exists a circular arc connecting c and v that lies entirely in S (see Fig. 8(a)).

Lemma 5 *Let S be a nonreflex spherical triangle and let u be a point in the interior of S. Then u is spherically visible from every of the three corners of S. Furthermore, the arc that witnesses the spherical visibility and the boundary arcs of the corresponding corner c have all a distinct tangent at c.*

Proof. Let the corners of S be v_1, v_2, v_3. We prove the statement for the corner v_1. Let f be the Möbius transformation that maps the arcs v_1v_2 and v_1v_3 to straight-line segments. Clearly, $u' := f(u)$ lies inside $S' := f(S)$. We set $v_i' := f(v_i)$. Let s be the ray that starts in v_1' and is pointed towards u'. Since Möbius transformations are conformal, the angles at v_2' and v_3' in S' are both less than π. It follows that s hits first the vertex u' and then the boundary of S' without reentering. Therefore, the Möbius function f^{-1} maps the segment $v_1'u'$ to a circular arc that witnesses the spherical visibility of u in S. Clearly, the tangents of v_1v_2, v_1v_3, and $f^{-1}(s)$ are all distinct in v_1 because f and f^{-1} are conformal. Fig. 8 shows an example of a triangle S and its image S'. □

Lemma 6 *Let G be a k-fan with outer face f_0, and let S be a nonreflex spherical triangle. Then G can be drawn with $k+4$ circular arcs such that the boundary of S realizes f_0.*

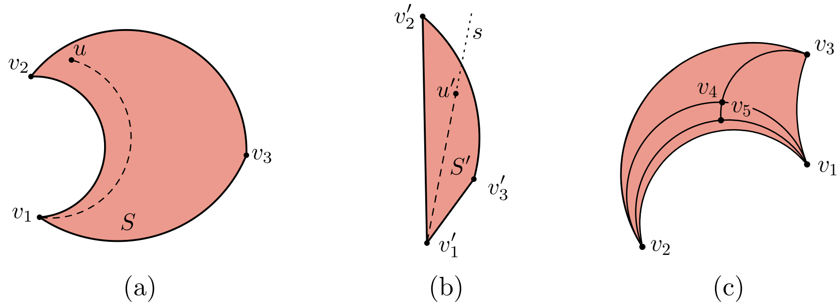


Figure 8: (a) A spherical triangle spanned by v_1, v_2, v_3 with interior point u. (b) The image of the same spherical triangle under a Möbius transformation that turns two boundary arcs into straight-lines. (c) Construction in the proof of Lemma 6.

Proof. Let us first discuss the case $k = 1$. Let the vertices of f_0 be v_1, v_2, and v_3. The vertex v_4 is placed reasonably close to the arc v_1v_2, such that the arc connecting v_1 with v_2 via v_4 lies inside S. By this we define a new spherical triangle $S' \subsetneq S$, which has the corners v_1, v_2 and v_3. Due to Lemma 5, v_4 is spherically visible from v_3 in S', and therefore we can connect v_3 with v_4 by an arc inside S'. The drawing needs five arcs.

Assume now that G is a 2-fan. We extend the arc ending at v_4 (without changing the curvature), such that it reaches inside the spherical triangle spanned by v_1, v_2, v_4. Let the endpoint of the extended arc be v_5. We can interpolate in between the two arcs between v_1 and v_2 such that we get a circular arc connecting v_1 and v_2 via v_5. The new arc does not introduce any crossings. See Fig. 8(c) for an illustration. By repeating this argument, we can draw every k-fan with $k + 4$ arcs. □

Theorem 3 *Every planar 3-tree $G = (V, E)$ can be drawn with $11|E|/18 + 3$ arcs as a composite arc-drawing. For every number n, there is a planar 3-tree $G = (V, E)$ with more than n vertices, whose composite arc-drawings require at least $|E|/6$ arcs.*

Proof. Note that we can naturally recurse on a planar 3-tree since when the first vertex v_4 is stacked on the face $v_1v_2v_3$, the graphs contained in the three interior triangles are planar 3-trees as well. For the drawing algorithm we assume that there were at least two stacking operations, otherwise the bound of the lemma follows directly. Let G_f the subgraph of G that is isomorphic to a k-fan and that includes the boundary face, such that $k \geq 2$ (see the solid-edge subgraph in Fig. 9). We draw G_f as discussed in Lemma 6 including all induced 1-fans of G that would lie inside faces of G_f. For every 1-fan we need 2 arcs. This implies that in the worst case there is such a 1-fan for every face in G_f, except for v_1, v_2, v_{k+3}. Therefore we have $2k$ 1-fans, contributing a total of $4k$ arcs.

The k-fan requires $k+1$ arcs for the interior edges. Thus we have $5k+1$ arcs for the $9k$ interior edges. This shows that the ratio between interior arcs and edges is at most $11/18$ (recall that $k \geq 2$).

For all faces of G_f that are not faces of G we still have to fill in a subgraph of G. These subgraphs (including its face in G_f) form planar 3-trees as well. We draw these 3-trees (recursively) with the above strategy. To do so, we draw the boundary face, which is an interior face of G_f, as it was drawn for G_f. By the above analysis we "save" for every subgraph a ratio of $11/18$ of the interior edges. The asserted bound of $11|E|/18+3$ follows.

The lower bound is due to Lemma 1, since an arbitrarily large planar 3-tree with odd degree vertices only can be easily constructed (see Fig. 9 for an example). □

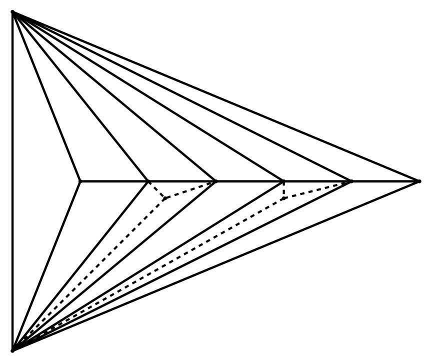
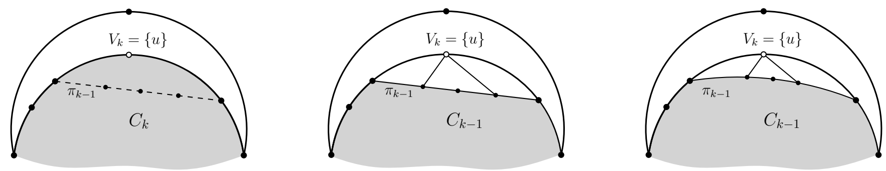


Figure 9: A planar 3-tree with odd degree vertices only.

4 Composite drawings of 3-connected planar graphs

Let $G=(V,E)$ be a planar 3-connected graph with n vertices. We order the vertices of G with respect to some *canonical order* as defined by Kant [9]. In particular, let $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$ a partition of the set V. The subgraph induced by $V_1 \cup \cdots \cup V_k$ is called G_k and the boundary face of G_k is callled C_k. The partition $\mathcal{V}$ is called a cannonical order, if

- $V_1 = \{v_1, v_2\}$ and $(v_1, v_2) \in E$,
- $V_m = \{v_n\}$ and $(v_1, v_n) \in E$,
- G_k is 2-connected and internally 3-connected,
- for each $k \in \{2, \ldots, m-1\}$ **either** $V_k = \{z\}$ where $z \in C_k$ and z has a neighbor in $\bigcup_{j>k} V_j$, **or** $V_k = \{z_1, \ldots z_t\}$ where each z_i has a neighbor



(a) (b) (c)

Figure 10: A step in the drawing algorithm for planar 3-connected graphs. First, the path π_{k-1} is added (a), If V_k was a singelton, possible edges in between π_{k-1} and C_k are drawn (b). After a perturbation the intermediate face C_{k-1} becomes convex.

in $\bigcup_{j>k} V_j$, and where z_1 and z_t have each one neighbor in C_{k-1} and no other vertex in V_k has a neighbor in G_{k-1}.

Every planar 3-connected graph has such a canonical order [9].

We use the reversed canonical order to draw the graph G. As an invariant we maintain that after processing V_{k+1} the face C_k is indeed a face in the preliminary drawing, and furthermore it encloses a convex region. The algorithm starts with realizing C_m as a circle and placing the vertices of C_m on that circle. Since we do not allow circles as geometric entites, this circle is understood as an arrangement of two circular arcs. In every step we process a set V_k of the canonical order, where we always pick the set with the highest index that has not been processed yet. We denote by E_{k-1} the edges on C_{k-1} that are not on C_k. The edges E_{k-1} form a (connected) subpath of C_{k-1} which we denote with π_{k-1}. When processing V_k we draw π_{k-1} together with all edges that connect the vertices of π_{k-1} with already drawn vertices. The new intermediate drawing is obtained in three steps (see Fig. 10).

1.) We draw π_{k-1} as a straight-line segment connecting its two endpoints which have to lie on C_k. This can be done without creating intersections since C_k is a convex face.

2.) We draw the edges between π_{k-1} and C_k. Note that unless $|V_k| = 1$ there are no such edges. Hence we can assume that $V_k = \{u\}$. Again we can use straight-line segments for the edges, since u can see all of π_{k-1}.

3.) We perturb the line segments representing π_{k-1} such that C_{k-1} becomes strictly convex. The edges drawn in the 2.) step will be updated accordingly.

Clearly, after updating the drawing the invariant holds. Fig. 11 shows an example.

Theorem 4 *The above method constructs a composite arc-drawing of a planar 3-connected graph $G = (V, E)$ with at most $2|E|/3$ arcs. For every n there is a*

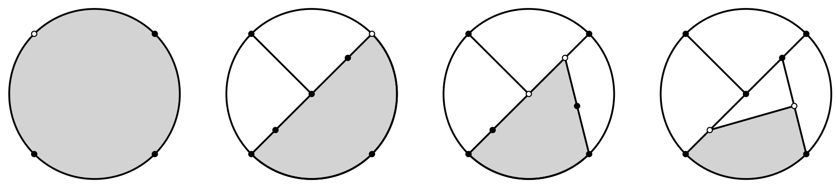


Figure 11: The graph of the cube drawn with the technique based on the reverse canonical order of the vertices. The faces C_k and the sets V_k are highlighted. The perturbation of the straight-line segments was not necessary in this example.

triangulation $G = (V, E)$ with more than n vertices whose composite segment-drawings need at least $|E|/6 + 1$ segments.

Proof. As a first step we count the number of arcs used in the drawing. We introduced three types of arcs. (i) Every path π_i will be realized as a single arc. Such an arc will be added $m - 2$ times (for every transition $V_{i+1} \to V_i, 2 \leq i \leq m - 1$). (ii) When processing a singleton set $V_i = \{u\}$ we might draw additional arcs for edges (u, u') that are not on π_i. We call these arcs *short arcs* and denote the number of short arcs by s. (iii) For the boundary face we need two arcs. In total we have thus $m + s$ arcs.

For the further analysis we introduce a function Φ who assigns every graph G_i a potential. Let the number of edges in G_i that are realized as short edges in the drawing be s_i, and the number of interior faces in G_i be f_i. Then we define $\Phi(G_i) := f_i - s_i$. Although we used the *reversed* canonical order in the drawing algorithm, we consider the forward canonical order for the analysis. Clearly, we have $\Phi(G_1) = 0$. When we add a vertex set in the canonical order the potential changes. In case we add a singleton, we introduce a number of faces, say f_{new}, but we also add $f_{\text{new}} - 1$ edges which will be drawn as short arcs. Hence, we get $\Phi(G_i) = \Phi(G_{i-1}) + 1$. In case our set is not a singleton set we add only one face, and no short arc edge. Again we have $\Phi(G_i) = \Phi(G_{i-1}) + 1$. As a consequence $\Phi(G) = m - 1$, and therefore $f - s = m - 1$, for f being the number of interior faces in G.

The number of arcs can now be expressed as $m+s = f+1$. The ratio between arcs and edges is maximized when a graph with $|E|$ edges has a maximal number of vertices, that is, when G is a triangulation and $|E| = 3|V| - 6$. By Euler's formula $f - |E| + |V| = 1$ we obtain

$$\begin{aligned}\#\text{ arcs} = f + 1 &= |E| - |V| + 2 \\ &= |E| - 1/3|E| - 2 + 2 \\ &= 2/3|E|,\end{aligned}$$

which proves the upper bound from the lemma. The lower bound follows from the lower bound of Theorem 3. □

Note that we have freedom how to place the vertices horizontally. The algorithm can be easily updated such that all vertices have distance at least one while keeping the largest distance between any two points at most n. From this perspective the drawing has a good vertex resolution.

5 Future work

In this paper we presented the first algorithms for composite drawings. For all graph classes except for trees there is a gap between the lower and upper bound on the number of necessary arcs. We are interested in tightening these gaps, but we think that new methods are required for a substantial improvement.

This paper concentrates on the combinatorial question, i.e., how small can the visual complexity be. On the other hand, drawings with very low visual complexity might violate other criteria for readable drawings. We addressed this issue in Theorem 1 by combining classical graph drawing criteria (grid size) with low visual complexity. We would like to extend this result for more complicated graph classes in order to construct more readable drawings with low visual complexity.

It is ongoing research to evaluate with empirical user studies our hypothesis that a graph with low visual complexity is easier to percept by the viewer. Our hope is that we can show that drawings with small visual complexity are easier to memorize and we think this might be especially applicable to drawings of graphs with a small number of vertices.

Finally, we would like to point out that we are interested in small decompositions of planar graphs into edge-disjoint simple paths. This graph-theoretic question might yield better lower bounds. Although this problem seems elementary, only partial results are known. If the graph is a triangulation, it can be decomposed into edge-disjoint simple paths that all have exactly three edges [8]. The same is true for cubic bridge-less graphs [4]. We would like to see a similar bound for general planar 3-connected graphs.

Acknowledgement

I thank Thomas van Dijk, Phillip Hanraths, Jan-Henrik Haunert, Wouter Meulemans, Birgit Vogtenhuber, and Alexander Wolff for helpful discussions on this subject.

References

[1] O. Aichholzer, W. Aigner, F. Aurenhammer, K. C. Dobiásová, B. Jüttler, and G. Rote. Triangulations with circular arcs. In M. J. van Kreveld and B. Speckmann, editors, *Graph Drawing - 19th International Symposium, GD 2011*, volume 7034 of *Lecture Notes in Computer Science*, pages 296–307. Springer, 2011. doi:10.1007/978-3-642-25878-7_29.

[2] M. J. Alam, M. A. Bekos, M. Kaufmann, P. Kindermann, S. G. Kobourov, and A. Wolff. Smooth orthogonal drawings of planar graphs. In A. Pardo and A. Viola, editors, *LATIN 2014: Theoretical Informatics - 11th Latin American Symposium*, volume 8392 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2014. doi:10.1007/978-3-642-54423-1_13.

[3] M. A. Bekos, M. Kaufmann, S. G. Kobourov, and A. Symvonis. Smooth orthogonal layouts. *J. Graph Algorithms Appl.*, 17(5):575–595, 2013. doi:10.7155/jgaa.00305.

[4] A. Bouchet and J.-L. Fouquet. Trois types de décompositions d'un graphe en chaînes. In C. Berge, D. Bresson, P. Camion, J. Maurras, and F. Sterboul, editors, *Combinatorial Mathematics Proceedings of the International Colloquium on Graph Theory and Combinatorics*, volume 75, pages 131–141. North-Holland, 1983. doi:10.1016/S0304-0208(08)73380-2.

[5] C. C. Cheng, C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Drawing planar graphs with circular arcs. *Discrete & Computational Geometry*, 25(3):405–418, 2001. doi:10.1007/s004540010080.

[6] V. Dujmović, D. Eppstein, M. Suderman, and D. R. Wood. Drawings of planar graphs with few slopes and segments. *Comput. Geom.*, 38(3):194–212, 2007. doi:10.1016/j.comgeo.2006.09.002.

[7] D. Eppstein. Parallel recognition of series-parallel graphs. *Inf. Comput.*, 98(1):41–55, 1992. doi:10.1016/0890-5401(92)90041-D.

[8] R. Häggkvist and R. Johansson. A note on edge-decompositions of planar graphs. *Discrete Mathematics*, 283(1-3):263–266, 2004. doi:10.1016/j.disc.2003.11.017.

[9] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16(1):4–32, 1996. doi:10.1007/BF02086606.

[10] M. Lombardi and R. Hobbs. *Mark Lombardi: Global Networks.* Independent Curators, 2003.

[11] G. L. Miller and V. Ramachandran. Efficient parallel ear decomposition with applications. unpublished manuscript, MSRI, Berkeley, CA., January 1986.

[12] H. C. Purchase, J. Hamer, M. Nöllenburg, and S. G. Kobourov. On the usability of lombardi graph drawings. In W. Didimo and M. Patrignani, editors, *Graph Drawing - 20th International Symposium, GD 2012*, volume 7704 of *Lecture Notes in Computer Science*, pages 451–462. Springer, 2012. doi:10.1007/978-3-642-36763-2_40.

[13] J. Sally and P. J. Sally. *Roots to research: a vertical development of mathematical problems*. American Mathematical Society (AMS), Rhode Island, USA, 2007.

[14] A. Schulz. Drawing graphs with few arcs. In A. Brandstädt, K. Jansen, and R. Reischuk, editors, *Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013*, volume 8165 of *Lecture Notes in Computer Science*, pages 406–417. Springer, 2013. doi:10.1007/978-3-642-45043-3_35.

[15] R. E. Tarjan. Linking and cutting trees. In *Data Structures and Network Algorithms*, chapter 5, pages 59–70. Society for Industrial and Applied Mathematics (SIAM), 1983.

[16] K. Xu, C. Rooney, P. Passmore, D.-H. Ham, and P. H. Nguyen. A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2449–2456, 2012. doi:10.1109/TVCG.2012.189.