# Drawing Graphs with Few Arcs⋆

André Schulz

Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, Germany. andre.schulz@uni-muenster.de

**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 ground set of arcs/segments for a composite arc/segment-drawing. We present algorithms for constructing composite arc-drawings 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.

## 1 Introduction

A graph is drawn by realizing its vertices as points in the plane and connecting adjacent vertices by continuous curves. There exists a large number of design criteria such as small area, high 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 viewer. In this paper we propose a novel criterion for aesthetic and readable graph drawings. Our goal is to generate drawings that are easy to *percieve* 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. In this paper we go beyond crossing-free straight-line drawings and try to reduce the number of geometric entities of a drawing further. To make this possible, we group edges, such that they form a new entity. For example, if we are able to

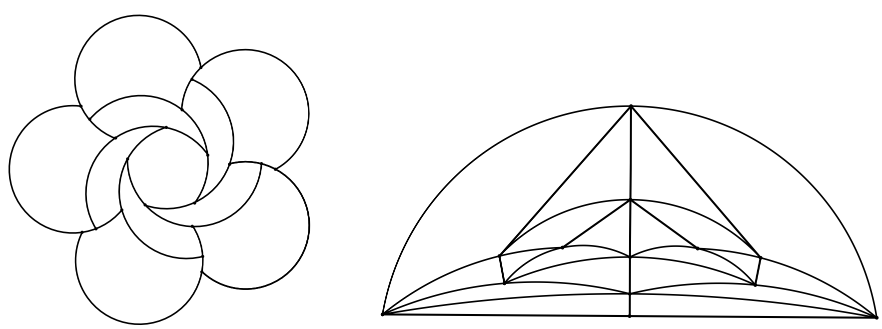
⋆ The research was performed in cooperation with Eurogiga project GraDR - Graph Drawings and Representations, and supported in part by ESF Eurogiga networking grant.

draw a path of the graph as straight-line 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 an example of a composite arc drawing that uses circular arcs as basic objects.

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 [9]. The drawings of Lombardi had 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. Moreover, by aligning vertices Lombardi included additional information in the drawings. For example, the alignment was used to visualize temporal or sequential dependencies of events represented by vertices.



(a) (b)

**Fig. 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).

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 is the most natural way for drawing edges, but also circular arcs have been proposed as "edge shapes" before [1,3]. In our understanding, a line segment is a

| graph class | upper bounds | | lower bounds | |
|---|---|---|---|---|
| | segments [5] | arcs | segments | |
| trees | $\lceil \lvert E\rvert/2\rceil$ | $\lceil \lvert E\rvert/2\rceil$ | $\lceil \lvert E\rvert/2\rceil$ | [5] |
| trees 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. Some lower bounds are presented as slightly simplified expressions.

degenerated circular arc, and by a suitable Möbius transformation these segments can be converted to circular arcs. We present bounds on the maximal number of arcs/segments necessary in a composite drawing. Our approach can not 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 (non-crossing) 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 observation.

**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 to start, respectively end. Hence we have at least $N$ endpoints of arcs. □

*Related Work.* Dujmović et al. [5] 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 imply that in a composite drawing circular arcs are indeed more powerful than segments, since they 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.

Recently, user studies comparing straight-line drawings with circular-arc drawing were conducted [11,15]. 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 [11]. 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.

## 2 Composite drawing 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 [5], 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 yields an embedding on a grid exponential in $O(|V|)$. Therefore, the produced drawing cannot be placed on a polynomial grid.

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 with only one circle. However realizing a circle such that it contains many grid points is a highly non-trivial task. To our knowledge the best method uses a grid of size $O(5^{n/4})$ [12].

**Heavy edge path-decomposition.** The drawing algorithm is based on a decomposition scheme for trees, called the *heavy edge path-decomposition* [13], 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 induced by a heavy path is the subtree rooted at its top node. The light edge that links the top node with its parent in $T$ is called *light parent edge*. 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.

**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*. 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 $3 \times 1$ 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 how to build the subtrees of the heavy paths (see Fig. 2(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. 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 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. 2(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. 2(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 vertically apart from $u$. 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 node $w$ following $u$ on the heavy path is placed at the bottom boundary of the row directly below $u$.

**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 $v_i$s 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

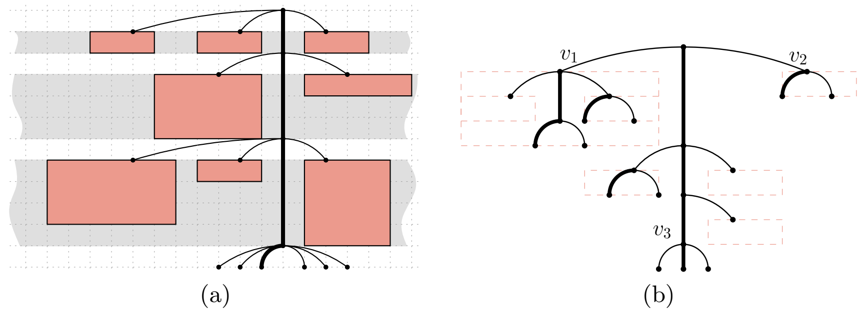



**Fig. 2.** (a) A drawing of a heavy path's subtree. Rows are drawn shaded and the heavy path is drawn thick. (b) An example of a composite arc-drawing of a tree. The safe boxes used in the algorithm are indicated by dashed rectangles. Vertex $v_1$ is a hep, $v_2$ is a 2-fork and, $v_3$ is a 3-fork.

$v_j$, and connect both by an arc running through $u$. This implies that $x_i = -x_j$ for every such pair of vertices.

We determine the location of the safe boxes by a greedy strategy (see Fig. 3). Let $\ell_i$ 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 $\ell_i$ 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 $B_{2t}$ 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 $b_i := r_i$.

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 $x = 0$ line. 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 $x_2$. 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* $\Delta$, that is the distance between the most extreme top corners. The only exception is when $k = 1$; in this case $\Delta$ equals the width of the only safe box plus 2.

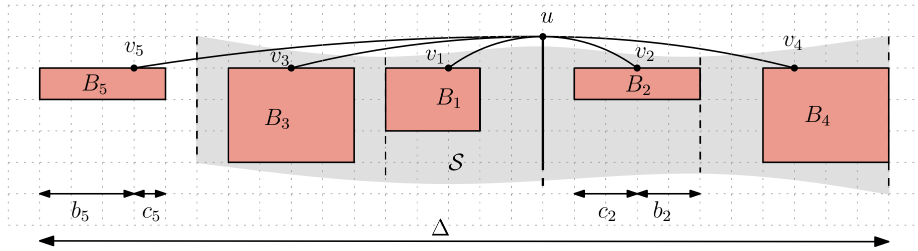



**Fig. 3.** A snapshot during the execution of the greedy strategy for the box displacement. The safe boxes up to $B_4$ have already been placed. When placing the last box $B_5$ we avoid the restricted strip $\mathcal{S}$. The boundaries of the strip $\mathcal{S}$ after round 1 and 2 are drawn as dashed lines.

**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*

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

The proof of the lemma can be found in the appendix. 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. 3 shows an example of the greedy strategy. This leads to the following lemma, whose proof can be found in the appendix.

**Lemma 3** *By inductively laying out the safe boxes with the greedy strategy explained above, the heavy edge 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 $B_1, \ldots, B_k$. 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 increase the height by $m + \sum_{i=1}^{k} h_i$. 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 edge path-decomposition the recursion depth is at most $\lceil \log n \rceil$. By induction a subtree of a heavy path with depth $k$ and $n'$ vertices is contained inside a safe box of width at most $3 \cdot (7/4)^k \cdot n'$. Hence the whole tree is contained in a box of width $3n \cdot (7/4)^{\lceil \log n \rceil}$ which is upper bounded by $O(n^{1.81})$. □

**Analysis.** A node is a *heavy even-prefork* (short *hep*), if its heavy edge child is a $k$-fork, with $k$ even. Fig. 2(b) illustrates the definitions. A charging scheme for the “saved edges” in forks and heps leads to the following lemma, whose proof can be found in the appendix. Combining Lemma 2, 3, 4 yields Theorem 1.

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

**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 2-trees and planar 3-trees

In this section we study composite arc-drawings of series-parallel graphs (also known as 2-trees) and planar 3-trees. We start with the series-parallel graphs.

Every series-parallel graph $G = (V, E)$ 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 [6]. Such a decomposition is called a *nested open ear-decomposition.* Based on the series-parallel composition history of $G$ a nested open ear-decomposition can be easily constructed.

**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 $|E|/4 + 1/2$ segments.*

The proof of the theorem can be found in the appendix.

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 $G = (\{v_1, \ldots, v_n\}, E)$ is a triangulation with facial structure, 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 *$k$-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 $(v_i, v_{i-1})$.

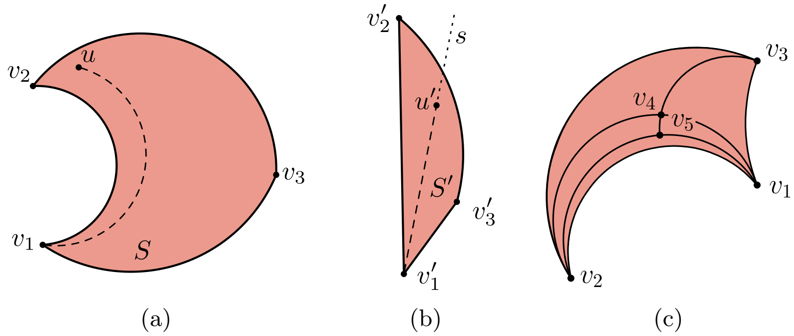



**Fig. 4.** (a) A spherical triangle spanned by $v_1, v_2, v_3$ with interior point $u$. (b) The image of the same spherical triangle under a Möbius transformation that turns two boundary arcs into straight-lines. (c) Construction in the proof of Lemma 6.

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 and every angle at a triangle corner is larger than zero and smaller than $\pi$. We say a vertex $v$ inside a spherical triangle $S$ is *spherically visible* from a triangle corner $c$, if there exists a circular arc connecting $c$ and $v$ that lies entirely in $S$ (see Fig. 4(a)).

**Lemma 5** *Let $S$ be a spherical triangle and let $u$ be a point inside $S$. 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 $c$, if $u$ is not on a boundary arc incident to $c$.*

**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-lines. 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 $u'$. The ray $s$ cannot intersect the arcs $v'_1v'_2$ and $v'_1v'_3$, since Möbius transformations are conformal and therefore the angles at $v'_2$ and $v'_3$ in $S'$ are both less than $\pi$. The ray $s$ has to intersect $S'$ an odd number of times. 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 $v'_1u'$ 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. 4 shows an example of a triangle $S$ and its image $S'$. □

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

**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. 4(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 $3 + 11|E|/18$ arcs as a composite arc-drawing. For every number $n$, there is a planar 3-tree $G = (V, E)$ with more then $n$ vertices, whose composite arc-drawings require at least $|E|/6$ arcs.*

The proof of the theorem can be found in the appendix.

## 4 Composite drawings of 3-connected planar graphs

Let $G = (V, E)$ be a triangulation. We order the vertices of $G$ with respect to some *canonical order* as defined by de Fraysseix, Pach, and Pollack [4]. In particular, let $v_1, v_2, \dots, v_n$ be the vertices of $G$, such that $D_i$ is the boundary face of the graph $G_i$ induced by $\{v_1, v_2, \dots v_i\}$. The graph $G_{i+1}$ is obtained by introducing the new vertex $v_{i+1}$ that is connected to some (at least 2) vertices of $D_i$. The boundary face $D_{i+1}$ is updated accordingly, also $D_2 = G_2$ is the initial segment.

The composite arc-drawing is constructed as follows. $G_2$ is drawn as segment of length 1. We now add vertex by vertex until we generated $G_n = G$. When a vertex, say $v_i$, is added, it is connected to some vertices $U$ in $D_{i-1}$. Let $u_\ell$ the leftmost and $u_r$ be the rightmost of these vertices relative to the edge $(v_1, v_2)$. We connect $u_\ell$ and $u_r$ by a circular arc $C_i$ with some radius $r_i$ and then place $v_i$ on this arc and connect $v_i$ to the remaining points in $U$ by straight-line segments. It might be the case that we could not succeed in finding the desired arc $C_i$. One reason might be that the angles at $u_\ell$ and $u_r$ on the current outer face do not support such an arc or the possible arcs would intersect $D_{i-1}$. It might also be that such an arc exists, but it is not possible to place $v_i$ such that it connects to all points in $U$ without creating an intersection. In both cases we start to redraw $G_{i-1}$ with a new set of radii $(r_j)_{j<i}$. All radii will be strictly enlarged. We start with an increased radius $r_3$. Then we continue along the canonical ordering and increase the radii accordingly. In every step we have increased the "angular space" that is available for the circular arcs on the current outer face. Also, if

some radii were determined by obstacles (arcs placed before), these obstacles became flatter, and also in this case the radii can be increased. Once we have increased the radii $(r_j)_{j<i}$ sufficiently enough to place arc $C_i$ we continue with the arc $C_{i+1}$ until we have constructed $G$. Fig. 1(b) shows a drawing constructed with the described strategy. If $G$ is a planar 3-connected graph, then we can find a similar construction based on the canonical ordering of Kant [8], that creates a composite drawing with at most $2|E|/3$ arcs.

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

**Proof.** The drawing obtained by the technique explained above draws for every vertex (except $v_1$,$v_2$) two edges as one arc. For a planar graph there are at most $3|V| - 6$ edges. Hence, the number of arcs differs from $|E|$ by $|V| - 2$, which shows the first statement of the theorem. The lower bound follows from the lower bound of Theorem 3. □

## 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 our hypothesis, that a graph with low visual complexity is easier to percept by the viewer, by empirical user studies. 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 for 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 [7]. The same is true for cubic bridge-less graphs [2]. 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, 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 van Kreveld and Speckmann [14], pages 296–307.
2. 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.
3. 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.
4. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
5. 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.
6. D. Eppstein. Parallel recognition of series-parallel graphs. *Inf. Comput.*, 98(1):41–55, 1992.
7. R. Häggkvist and R. Johansson. A note on edge-decompositions of planar graphs. *Discrete Mathematics*, 283(1-3):263–266, 2004.
8. G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16(1):4–32, 1996.
9. M. Lombardi and R. Hobbs. *Mark Lombardi: Global Networks*. Independent Curators, 2003.
10. G. L. Miller and V. Ramachandran. Efficient parallel ear decomposition with applications. unpublished manuscript, MSRI, Berkeley, CA., January 1986.
11. H. C. Purchase, J. Hamer, M. Nöllenburg, and S. G. Kobourov. On the usability of Lombardi graph drawings. In *Graph Drawing - 20th International Symposium, GD 2012, Redmond, Washington*, 2012.
12. J. Sally and P. J. Sally. *Roots to research: a vertical development of mathematical problems*. American Mathematical Society, Rhode Island, USA, 2007.
13. R. E. Tarjan. Linking and cutting trees. In *Data Structures and Network Algorithms*, chapter 5, pages 59–70. Society for Industrial and Applied Mathematics, 1983.
14. M. J. van Kreveld and B. Speckmann, editors. *Graph Drawing - 19th International Symposium, GD 2011, Eindhoven, The Netherlands, September 21-23, 2011, Revised Selected Papers*, volume 7034 of *Lecture Notes in Computer Science*. Springer, 2012.
15. 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.

# A Omitted proofs

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

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

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

Let us first consider the case where $k$ 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 $\Delta$ 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

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

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 $b_{k-2} \leq X_b/3$.

If $k$ is odd 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

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

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.

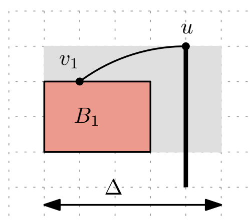



**Fig. 5.** The special case where $u$ has only one child connected by a light edge.

We finish the proof by looking at the special case where $k = 1$. In order to maintain the invariant, that the root does not lie on a corner of the safe box, we have to extend the width $\varDelta$ (see Fig. 5). We obtain $\varDelta = 2 + w_1$, which is smaller than $7/4w_1$ for $w_1 \ge 3$. Since we placed all leaves inside $3 \times 1$ boxes the lemma also holds for this extreme case. □

**Lemma 4** *Let $T = (V, E)$ be a tree drawn as a composite arc-drawing with the algorithm based on the heavy edge 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 edges incident to a $k$-fork. We define $E_{\text{odd}} := \{E_k \mid k \text{ odd}\}$ and $E_{\text{even}} := E \setminus E_{\text{odd}}$. The edges for all forks are disjoint and hence we have $|E| = |E_{\text{odd}}| + |E_{\text{even}}| \leq \sum_k |E_k|$. 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, we have

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

If $S$ is a single node 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

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

For every set $E_k$ we have that $|E_k|/(k+1) = |F_k|$. 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 (1) and observe 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 $T$. □

**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 $|E|/4 + 1/2$ segments.*

**Proof.** Let $E_1, E_2, \ldots E_k$ be a nested open ear-decomposition of $G$. As noted by Miller and Ramachandran [10],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 $E_j$. 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 decreasing 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. 6(a) shows a drawing obtained by the algorithm.

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

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

For the lower bound consider the graph which alternatively joins a single line by a series and parallel composition (see Fig. 6(b)). Lemma 1 implies that such a graph needs at least $|E|/4 + 1/2$ arcs in any composite arc-drawing. □

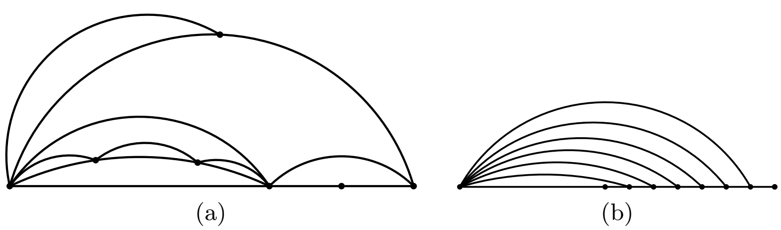



**Fig. 6.** (a) A composite arc-drawing of a series-parallel graph, obtained by the method explained in the proof of Theorem 2. (b) A series-parallel graph with only one even-degree vertex.

**Theorem 3** *Every planar 3-tree $G = (V, E)$ can be drawn with $3 + 11|E|/18$ arcs as a composite arc-drawing. For every number $n$, there is a planar 3-tree $G = (V, E)$ with more then $n$ 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 2 stacking operations. Let $G_f$ the subgraph of $G$ that is isomorphic to a $k$-fan and that includes the boundary face, such that $k$ is maximal (see the solid-edge subgraph in Fig. 7). We draw $G$ as discussed in

Lemma 6 including all induced 1-fans of $G$ that would lie inside faces of $G_f$. For every such 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 $k \geq 2$). The faces of $G_f$ that contain parts of $G$ with more than one additional vertex are analyzed by recursion. The asserted bound of $3 + 11|E|/18$ 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. 7 for an example). □

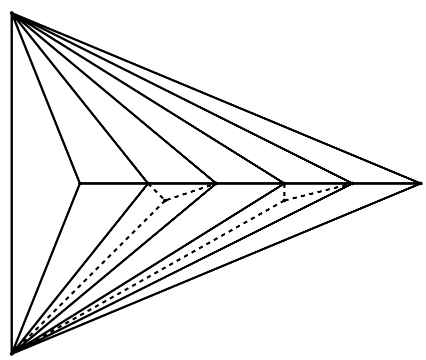


**Fig. 7.** A planar 3-tree with odd degree vertices only.