A Radial Adaption of the Sugiyama Framework for Hierarchical Graph Drawing

Technical Report MIP-0603
April, 2006

Christian Bachmaier and Michael Forster

University of Passau, Germany
94030 Passau, Germany
Fax: +49 851 509 3032
`{bachmaier,forster}@fmi.uni-passau.de`

Abstract. In radial drawings of hierarchical graphs the vertices are drawn on concentric circles instead of on horizontal lines as in the standard Sugiyama framework. This drawing style is well suited for the visualisation of centrality in social networks and similar concepts. Radial drawings also allow a more flexible edge routing than horizontal drawings, as edges can be routed around the center in two directions. In experimental results this reduces the number of crossings by approx. 30% on average.
This paper is the last step to complete the framework for drawing hierarchical graphs in a radial fashion. We present three heuristics for crossing reduction in radial level drawings of hierarchical graphs, and also briefly cover extensions of the level assignment step to take advantage of the increasing perimeter of the circles.

1 Introduction

In hierarchical graph layout vertices are usually drawn on parallel horizontal lines, and edges are drawn as y-monotone polylines that may bend when they intersect a level line. The standard drawing algorithm [16] consists of four phases: *cycle removal* (reverses appropriate edges to eliminate cycles), *level assignment* (assigns vertices to levels and introduces dummy vertices to represent edge bends), *crossing reduction* (permutes vertices on the levels), and *coordinate assignment* (assigns x-coordinates to vertices, y-coordinates are implicit through the levels). See [11] for an extended overview.

The novelty of radial coordinate assignment [2] is to draw the level lines as concentric circles instead of as parallel horizontal lines. The apparent advantage is that level graphs can be drawn with fewer edge crossings. It is also more likely that a graph can be drawn without any crossings at all, i. e., the set of level planar graphs is a proper subset of the set of radial level planar graphs [1]. Note that radial level drawings are different from circular drawings [3, 13, 15] where only one circle contains all vertices. Here "inner level edges" with both end vertices on a common level are prohibited.

Radial level drawings are common, e. g., in the study of social networks [4,5]. There vertices model *actors* and edges represent *relations* between the actors. The importance (*centrality*) of a vertex is expressed by its distance (*closeness*) to the center, i. e., a position on a low level. Radial level drawings are also well suited for level graphs with an increasing number of vertices on higher levels. For example, in a graph that shows which web pages are reachable from a given start page by following k hyperlinks, higher levels are likely to contain many vertices while there are only few vertices on the lower levels.

In this paper we complete the Sugiyama framework for radial level drawings of hierarchical graphs by introducing methods for radial level assignment in Sect. 3 and radial crossing reduction in Sect. 4. The cycle removal does not differ from the horizontal case, thus standard algorithms can be used, see [11] for an overview. For the final coordinate assignment see [2].

2 Preliminaries

A *k-level graph* $G = (V, E, \phi)$ is a directed acyclic graph (DAG) with a level assignment $\phi\colon V \to \{1, 2, \ldots, k\}$, which partitions the vertex set into $k \leq |V|$ pairwise disjoint subsets $V = V_1 \dot\cup V_2 \dot\cup \cdots \dot\cup V_k$, $V_i = \phi^{-1}(i)$, $1 \leq i \leq k$, such that $\phi(u) < \phi(v)$ for each edge $(u, v) \in E$. Particularly, $k = 1$ implies that $E = \emptyset$. An edge (u, v) is *short* if $\phi(v) - \phi(u) = 1$, otherwise it is *long* and *spans* the levels $\phi(u) + 1, \ldots, \phi(v) - 1$. A level graph without long edges is *proper*.

An *ordering* of a proper level graph is a partial order $\prec$ of V such that $u \prec v$ or $v \prec u$ iff $\phi(u) = \phi(v)$. This is equivalent to a definition of the vertex positions on level i as a bijective function $\pi_i\colon V_i \to \{0, \ldots, |V_i| - 1\}$ with $u \prec v \Leftrightarrow \pi_i(u) < \pi_i(v)$ for any two vertices $u, v \in V_i$ and $\pi = (\pi_i)_{1 \leq i \leq k}$. We call an ordering of a level graph a *horizontal embedding*. Throughout the paper let $N^-(v) = \{\, u \mid (u, v) \in E \,\}$ denote the predecessors of $v \in V$ and $\operatorname{sgn}\colon \mathbb{R} \to \{-1, 0, 1\}$ be the signum function.

3 Radial Level Assignment

Although the main focus of this paper is on crossing reduction, it is also interesting to study the level assignment step for radial level drawings. The basic problem is the same as in horizontal level drawings: A given DAG is to be transformed into a level graph by assigning the vertices to levels. Thus, any existing level assignment algorithm for horizontal level drawings can directly be used for radial level drawings. The optimization criteria, however, slightly change: Radial level drawings use k concentric circles to place the vertices of the k levels. Contrary to the constant line lengths in horizontal level drawings, the perimeters of the circles get longer with ascending level numbers: On an outer circle, there is space for more vertices than on an inner circle. In the following, we investigate how level assignment methods can be extended to take advantage of this.

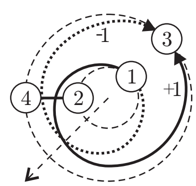
A straight-forward idea is to apply the longest path level assignment method from outer to inner levels: First, each sink of the graph is assigned to the highest level. For the remaining vertices the level is recursively defined by $\phi(v) = \min\{\, \phi(w) \mid (v, w) \in E \,\} - 1$. This puts each vertex on the outermost possible level while minimizing the number k of levels. There is no explicit balancing of level sizes, however.

For a better vertex distribution, an extension of the Coffman/Graham algorithm [6] can be used that explicitly takes into account the growing perimeter of the circles. The original Coffman/Graham algorithm computes a leveling where the number of vertices per level is bounded by a given constant W. We change this bound to a function $w(i)$ which grows proportionally to the number i of the level: $w(i) = W \cdot i$. The first phase of the algorithm remains unchanged, but we apply it using the opposite edge direction: After removing transitive edges in linear time, an appropriate ordering $o\colon V \to \{0, \dots, |V| - 1\}$ of the vertices is computed: Initially, all vertices are unnumbered. We consecutively choose one vertex at a time and assign the next ascending number to it. The vertex is chosen so that it has no unnumbered successors and that the numbers of the successors are minimal regarding a specific ordering of integer sets: a set of vertex numbers is considered less than another one, if the maximum is less. If the maximum of both sets is equal, the next smaller value is compared, and so on. In the second phase the algorithm places one vertex at a time, starting with the vertex numbered with $|V| - 1$ on level $i = 1$ and filling the levels from inner to outer circles. In one step it places the next unleveled vertex $v \in V$ with maximum $o(v)$ whose predecessors are already leveled. If level i is full, i. e., if i contains $w(i)$ vertices, or if v has a predecessor u with $\phi(u) = i$, then a new level is started, i. e., i is increased by 1. The level of v is then set to $\phi(v) = i$.

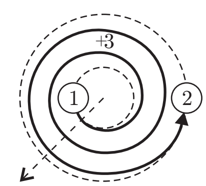
As the last step of the level assignment phase the level graph is made proper, because for drawing level graphs it is necessary to know where long edges should be routed, i. e., between which two vertices on a spanned level. Thus, each long edge $(u, v) \in E$ is subdivided by new *dummy vertices* v_i on the spanned levels $i = \phi(u) + 1, \dots, \phi(v) - 1$, $\phi(v_i) = i$. In total, this results in up to $\mathcal{O}(k|E|)$ dummy vertices and running time. In the following we will only consider proper level graphs.

4 Radial Crossing Reduction

Regardless of whether the leveling of a level graph is given by the application or if it has been computed by one of the algorithms in the previous section, the next step towards a hierarchical drawing is to compute an embedding. In horizontal hierarchical drawings the embedding is fully defined by the vertex ordering $\pi = (\pi_i)_{1 \le i \le k}$. For radial embeddings it is also necessary to know where the orderings start and end on each level. Therefore, we introduce a *ray* that tags this borderline between the vertices. The ray is a straight halfline from the center to infinity between the vertices on each level with extremal positions. Edges crossed by the ray are called *cut edges*. In horizontal drawings of level graphs a crossing between two edges only depends on the orderings of the end vertices. In radial level drawings, however, it is also necessary to consider the direction, in which the edges are wound around the center. There a two directions, clockwise and counter-clockwise, and edges can also be wound around the center multiple times. We call this the offset $\psi\colon E \to \mathbb{Z}$ of an edge. Thereby, $|\psi(e)|$ counts the crossings of an edge $e \in E$ with the ray. If $\psi(e) < 0$ ($\psi(e) > 0$), e is a *clockwise* (*counter-clockwise*) cut edge, i. e., the sign of $\psi(e)$ reflects the mathematical direction of rotation, see Fig. 1. If $\psi(e) = 0$, then e is not a cut edge and thus needs no direction information. Observe that a cut edge cannot cross the ray clockwise and counter-clockwise simultaneously. We define a *radial embedding* $\mathcal{E}$ of a graph $G = (V, E, \phi)$ to consist of the vertex ordering π and of the edge offsets ψ, i. e., $\mathcal{E} = (\pi, \psi)$.



(a) Edge $(1, 3)$ drawn counter-clockwise and clockwise (dotted)

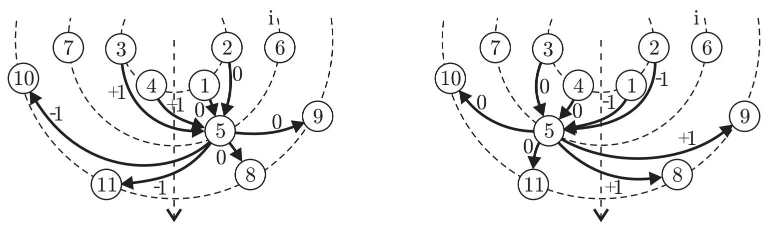


(b) $\psi((1, 2)) = +3$

Fig. 1. Offsets of edges

Compared to horizontal drawings there is an additional freedom in radial drawings without changing the crossing number: *rotation* of a level i. A clockwise rotation moves the vertex v with minimum position on the ordered level $\phi(v) = i$ over the ray by setting $\pi_i(v)$ to the maximum on i. The other values of π_i are updated accordingly. For an illustration see Fig. 2, where $v = 5$. A counter-clockwise rotation is symmetric. Rotations do not modify the "cyclic order", i. e., the neighborhood of every vertex on its radial level line is preserved. However, the offsets of the edges incident to v must be updated. If rotating clockwise, the offsets of incoming edges of v are reduced by 1 and the offsets of outgoing edges are increased by 1. The offset updates for rotating counter-clockwise are symmetric. Depending on the implementation, rotation needs $\mathcal{O}(\deg(v))$ resp. $\mathcal{O}(|V| + \deg(v))$ running time.

The most common technique for crossing reduction in proper level graphs is to only consider two consecutive levels at a time in multiple top-down and bottom-up passes. Starting with an arbitrary permutation of the first level, subsequently the ordering of one level is fixed, while the next one is reordered to minimize the number of crossings in-between. This *one-sided two-level crossing reduction problem* is $\mathcal{NP}$-hard [9] and well-studied. For radial level embeddings we follow the same strategy and consider the *radial one-sided two-level crossing reduction*



(a) Radial embedding (b) Clockwise rotation of level i

Fig. 2. Rotation

problem. Given a 2-level graph $G = (V_1 \dot\cup V_2, E, \phi)$ and an ordering π_1 of the first level, our objective is to compute an ordering of the second level and offsets for the edges with few crossings.

4.1 Properties

Crossings between edges in radial embeddings depend on their offsets and on the order of the end vertices. There can be more than one crossing between two edges, if they have very different offsets. We denote the number of crossings between two edges $e_1, e_2 \in E$ in an embedding $\mathcal{E}$ by $\chi_{\mathcal{E}}(e_1, e_2)$. Intuitively, this number is approximately equal to the difference of the offsets $|\psi(e_2) - \psi(e_1)|$. The exact formula is slightly different, however, with a small shift depending on the vertex ordering, see Lemma 1. The (radial) crossing number of a radial embedding $\mathcal{E}$ and of a level graph $G = (V, E, \phi)$ are then naturally defined as $\chi(\mathcal{E}) = \sum_{\{e_1,e_2\} \subseteq E, e_1 \neq e_2} \chi_{\mathcal{E}}(e_1, e_2)$ and $\chi(G) = \min\{\chi(\mathcal{E}) \mid \mathcal{E}$ is a radial embedding of $G\}$.

Lemma 1. *Let $\mathcal{E} = (\pi, \psi)$ be a radial embedding of a 2-level graph $G = (V_1 \dot\cup V_2, E, \phi)$. Then the number of crossings between two edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ is*

$$\chi_{\mathcal{E}}(e_1, e_2) = \max\left\{0, \left|\psi(e_2) - \psi(e_1) + \tfrac{b-a}{2}\right| + \tfrac{|a|+|b|}{2} - 1\right\} \quad , \textit{ where}$$
$$a = \operatorname{sgn}(\pi_1(u_2) - \pi_1(u_1)) \textit{ and}$$
$$b = \operatorname{sgn}(\pi_2(v_2) - \pi_2(v_1)) \quad .$$

Proof. In analogy to horizontal embeddings, edge crossings do not depend on the exact position of the end vertices, but only on the relative ordering ($\prec$, $\succ$, or $=$) and on the edge offsets. We can assume w. l. o. g. that $\psi(e_1) = 0$, because in any embedding we can rotate the whole second level multiple times without changing π_2 or the offset difference $\delta = \psi(e_2) - \psi(e_1)$. This leads to $3 \cdot 3 \cdot 3 = 27$ cases, which are straight-forward to prove, see Tab. 4.1. □

Table 1. The crossing number in relation to $\delta = \psi(e_2) - \psi(e_1)$

u_1 vs. u_2	v_1 vs. v_2	$\psi(e_2)$ vs. $\psi(e_1)$	$\chi_{\mathcal{E}}(e_1, e_2)$	u_1 vs. u_2	v_1 vs. v_2	$\psi(e_2)$ vs. $\psi(e_1)$	$\chi_{\mathcal{E}}(e_1, e_2)$
$\prec$	$\prec$	$<$	$\lvert\delta\rvert$	$\succ$	$\succ$	$>$	$\lvert\delta\rvert$
$\prec$	$\prec$	$=$	0	$\succ$	$\succ$	$=$	0
$\prec$	$\prec$	$>$	$\lvert\delta\rvert$	$\succ$	$\succ$	$<$	$\lvert\delta\rvert$
$\prec$	$=$	$<$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$	$\succ$	$=$	$>$	$\left\lvert\delta + \frac{1}{2}\right\rvert - \frac{1}{2}$
$\prec$	$=$	$=$	0	$\succ$	$=$	$=$	0
$\prec$	$=$	$>$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$	$\succ$	$=$	$<$	$\left\lvert\delta + \frac{1}{2}\right\rvert - \frac{1}{2}$
$\prec$	$\succ$	$<$	$\lvert\delta - 1\rvert$	$\succ$	$\prec$	$>$	$\lvert\delta\rvert + 1$
$\prec$	$\succ$	$=$	1	$\succ$	$\prec$	$=$	1
$\prec$	$\succ$	$>$	$\lvert\delta - 1\rvert$	$\succ$	$\prec$	$<$	$\lvert\delta\rvert + 1$
$=$	$\prec$	$<$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$	$=$	$\succ$	$>$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$
$=$	$\prec$	$=$	0	$=$	$\succ$	$=$	0
$=$	$\prec$	$>$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$	$=$	$\succ$	$<$	$\left\lvert\delta - \frac{1}{2}\right\rvert - \frac{1}{2}$
$=$	$=$	$<$	$\lvert\delta\rvert - 1$	$=$	$=$	$>$	$\lvert\delta\rvert - 1$
$=$	$=$	$=$	0				

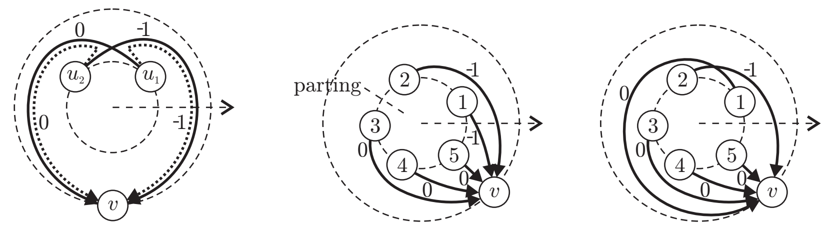
Corollary 1. *Let $\mathcal{E}$ be a radial embedding of a 2-level graph $G = (V_1 \dot{\cup} V_2, E, \phi)$. Swapping the position of two vertices $v, w \in V_2$ changes the number of crossing between two edges $(\cdot, v), (\cdot, w) \in E$ by at most 1.*

Before we show our radial crossing reduction algorithms, we first discuss some more properties which follow from radial level lines.

Lemma 2. *Let $G = (V_1 \dot{\cup} V_2, E, \phi)$ be a 2-level graph and let $e_1 = (u_1, v) \in E$ and $e_2 = (u_2, v) \in E$ be two edges with a common target vertex v. Then in any crossing minimal radial embedding $\mathcal{E} = (\pi, \psi)$ of G, $\pi_1(u_1) < \pi_1(u_2)$ implies $\psi(e_2) - \psi(e_1) \in \{0, 1\}$.*

Proof. Assume that $\psi(e_2) - \psi(e_1) \notin \{0, 1\}$. Then Lemma 1 implies $\chi_{\mathcal{E}}(e_1, e_2) > 0$. We choose an arbitrary crossing between e_1 and e_2 and show how the embedding can be modified to reduce the number of crossings, see Fig. 3(a) for an illustration. We exchange the routing of e_1 and e_2 between v and the crossing: e_1 is routed along the old course of e_2 until it reaches the crossing. The routing from there to u_1 is not changed. We symmetrically do the same with e_2. In the new embedding e_1 and e_2 have one crossing less and the number of crossings has not changed otherwise, contradicting the assumption and proving the lemma. □

Because of this result, it is clear that only embeddings need to be considered, where there is a clear *parting* between all edges incident to the same vertex as in Fig. 3(b). The parting is that position of the edge list of v that separates the two subsequences with offsets ψ_0 resp. $\psi_0 + 1$. Otherwise unnecessary crossings are generated between the incident edges, see Fig. 3(c). We also only consider radial embeddings with small edge offsets, because large offsets correspond to very long edges which are difficult to follow, and also lead to more crossings. Therefore, only the offsets −1, 0, and 1 are used in our algorithms.



(a) Sketch (b) With parting (c) Mixed

Fig. 3. Not all offset combinations for edges $(\cdot, v) \in E$ result in few crossings

Lemma 3. *Radial one-sided two-level crossing minimization is $\mathcal{NP}$-hard.*

Proof. We show the $\mathcal{NP}$-hardness by reduction from the horizontal one-sided two-level crossing minimization problem, which is known to be $\mathcal{NP}$-hard [9]. Given a 2-level graph $G = (V_1 \dot\cup V_2, E, \phi)$ with a fixed permutation π_1 of the first level, we construct a new 2-level graph $G' = (V_1' \dot\cup V_2', E', \phi')$ as follows: G is extended by $|E|^2$ new vertices $x_0, \dots, x_{|E|^2-1}$ at the end of the first level $\pi_1'(x_i) = |V_1| + i$, $\phi'(x_i) = 1$ and a new vertex y on the second level $\phi'(y) = 2$ that is connected to them by new edges $e_0, \dots, e_{|E|^2-1}$, $e_i = (x_i, y)$.

Let $\mathcal{E}' = ((\pi_1', \pi_2'), \psi)$ be a radial embedding of G' that has a minimum number of crossings subject to π_1'. We can assume w. l. o. g. (because of rotation and Lemma 2) that $\pi_2'(y) = |V_2|$ and $\psi(e_i) = 0$ for all new edges. Then none of the new edges has a crossing with any of the original edges, because this would lead to $|E|^2$ crossings, contradicting the minimality of the embedding. Thus, there are no cut edges, and $\pi_2 = \pi_2'|_{V_2}$ is a solution of the original horizontal one-sided two-level crossing minimization problem. □

As a consequence, we use heuristics for an efficient solution of the problem. In the following, we present three different approaches, extending some well-known horizontal one-sided two-level crossing reduction methods, namely the median, barycenter, and sifting heuristics.

4.2 Cartesian Barycenter

With some restrictions, the horizontal barycenter crossing reduction method can be directly used to compute a radial embedding: The horizontal vertex ordering defines a radial vertex ordering, and all edge offsets are set to 0. This does not use the additional freedom of radial edge routing, however, and therefore introduces more crossings than necessary. The result is especially bad for vertices whose neighbors on the first level are far apart. If, for example, a vertex is only adjacent to the first and last vertex of the first level, its best position is obviously near the ray, labeling one of the edges as a cut edge. But the horizontal algorithm cannot do that, and therefore produces an out-of-balance embedding. Even worse, the result depends on the current position of the ray.

One approach to improve that could be to rotate the first level before computing the barycenter values to an appropriate position, or maybe even use different rotations for different vertices. We propose a simpler, yet equally promising method. The basic idea stays the same: each vertex should be close to the average position of its neighbors. However, we use the terms “average” and “position” in a geometric sense. We assume the vertices of the first level V_1 to be uniformly distributed on a unit circle, according to the given ordering π_1. This defines Cartesian coordinates $(x(u), y(u)) \in \mathbb{R}^2$ for each $u \in V_1$. Then we compute for each $v \in V_2$ the *Cartesian barycenter*[1]

$$\mathrm{bary}(v) = \left(\frac{\sum_{u \in N^-(v)} x(u)}{|N^-(v)|}, \frac{\sum_{u \in N^-(v)} y(u)}{|N^-(v)|} \right)$$

of its predecessors $N^-(v)$ and sort the vertices circularly around the origin, i. e., by the angles of $\mathrm{bary}(v)$ in polar coordinates,

$$\beta(v) = \arctan \frac{y\left(\mathrm{bary}(v)\right)}{x\left(\mathrm{bary}(v)\right)} + \pi \cdot H\left(-x\left(\mathrm{bary}(v)\right)\right) \cdot \mathrm{sgn}\left(y\left(\mathrm{bary}(v)\right)\right) \quad ,$$

where $H(x) = 0$ for $x \leq 0$ and $H(x) = 1$ for $x > 0$ is the unit step function. Many programming languages provide a specialized function `atan2(x, y)` for this purpose.

After sorting, we distribute the vertices of the second level uniformly on a concentric circle with radius 2 and choose for the offset of each edge one of -1, 0, or $+1$, whichever leads to the shortest edge in a geometric sense. Obviously, this algorithm has the same running time as its horizontal version:

Theorem 1. *The running time of the Cartesian barycenter heuristic is* $\mathcal{O}(|E| + |V| \log |V|)$.

4.3 Cartesian Median

The Cartesian median heuristic is similar to the Cartesian barycenter heuristic. The only difference is that we take component-wise the x and y median instead of the component-wise barycenter. The running time stays the same, since $\mathrm{med}(v)$ can be computed in $\mathcal{O}(N^-(v))$, see [7]. The median values depend on the underlying coordinate system (origin and rotation). But since we use the same coordinates for all median computations, this is no problem. Rotated coordinate systems, however, might lead to a different results.

[1] Note that the division by $|N^-(v)|$ can be omitted in an implementation, because it does not change the polar angle of $\mathrm{bary}(v)$.

4.4 Radial Sifting

As a contrast to the fast and simple algorithms described above, we also developed an extension of the sifting heuristic, which is slower but generates less crossings. Sifting was originally introduced as a heuristic for vertex minimization in ordered binary decision diagrams [14] and later adapted for the (horizontal) one-sided crossing minimization problem [12]. The idea is to keep track of the objective function while moving a vertex $v \in V_2$ along a fixed ordering of all other vertices in V_2. Then v is placed to its locally optimal position. The method is thus an extension of the greedy-switch heuristic [8], where v is swapped iteratively with its successor. For crossing reduction the objective function is the number of crossings between the edges incident to the vertex under consideration and all other edges.

The efficient computation of crossing numbers in sifting for horizontal embeddings is based on the *crossing matrix*. Its entries correspond to the number of crossings caused by pairs of vertices in a particular relative ordering and can be computed as a preprocessing step. Whenever a vertex is placed to a new position, only a smallish number of updates is necessary. For radial embeddings, however, the crossings matrix cannot be computed in advance, because two vertices cannot be said to be in a particular (linear) relative order on radial levels.

Let $\mathcal{E} = (\pi, \psi)$ and $\mathcal{E}' = (\pi', \psi)$ be two embeddings of G, where $\mathcal{E}'$ is computed from $\mathcal{E}$ by swapping the vertex $v \in V_2$ and its successor $w \in V_2$ according to π_2, i. e., $\pi'_2(w) = \pi_2(v)$ and $\pi'_2(v) = \pi_2(v) + 1$. Since swapping the positions of v and w only affects the crossings of incident edges, the number of crossings in $\mathcal{E}'$ is efficiently computed as

$$\chi(\mathcal{E}') = \chi(\mathcal{E}) - c_{\mathcal{E}}(v, w) + c_{\mathcal{E}'}(v, w) \ , \text{ where}$$
$$c_{\mathcal{E}}(v, w) = \sum_{u \in N^-(v)} \sum_{x \in N^-(w)} \chi_{\mathcal{E}}\left((u, v), (x, w)\right) \ .$$

Unfortunately, we cannot directly transfer the ideas of [3] for the efficient computation of that formula, because in radial sifting the crossing numbers also depend on the edge offsets, which are not constant in our approach. A change in the offset of an edge may affect all other edges. Therefore, the overall running time of this part of the algorithm for one sifting round is $\mathcal{O}(|E|^2)$ instead of $\mathcal{O}(|V||E|)$. The total running time of the algorithm, however, is dominated by the next step, anyway.

In addition to the position of v, we also have to compute the offsets of the incident edges. As v moves along the second level circle in counter-clockwise direction, we update the offsets accordingly. Because of Lemma 2 we do not consider each possible offset combination for each position of v. Intuitively, the parting of the edges should move around the first level circle in the same direction as v, but on the opposite side of the circle. Otherwise, the edges incident to v get longer and tend to increase the number of crossings. Thus, we only decrease edge offsets by 1, starting with $\psi(e) = 1$ for all incident edges e, and we also do this one by one in the order of the end vertices on level 1. The decision for which offsets are updated at which position of v is made subject to whether this leads to an improvement or not. Note that the parting may move around level 1 twice, as offsets are decreased from 1 to -1.

Algorithm 1 shows one round of radial sifting, i. e., each vertex $v \in V_2$ is consecutively tested on each position once. We do not try the position $|V_2| - 1$, because it is equivalent to position 0 modulo rotation.

Theorem 2. *Given a 2-level graph $G = (V, E, \phi)$, the algorithm RADIAL-SIFTING runs in $\mathcal{O}(|V|^2 \cdot |E|)$ time.*

Proof. For each node $v \in V_2$ the content of the repeat-until loop in lines 13–26 is executed $\mathcal{O}(|V| + \deg(v))$ times: once per position, and additionally once per shifted parting. It is thus executed $\mathcal{O}(|V|^2 + |E|)$ times in total. As the running times of lines 15 and 17 are $\mathcal{O}(|E|)$, the repeat-until loop contributes $\mathcal{O}(|V|^2 \cdot |E| + |E|^2)$ to the overall running time.

The only other relevant part are lines 31 and 33, which are executed once for each pair (v, v_{i+1}). Since the summation needs $\mathcal{O}(\deg(v) \cdot \deg(v_{i+1}))$, the total running time of this part is $\mathcal{O}(|E|^2)$ and is therefore dominated by the above. □

To allow a harmonic drawing of the computed embedding in the next phase a final postprocessing step which rotates level 2 with respect to uniform edge lengths is useful. However, this is only for esthetic reasons and does neither affect the number of crossings nor the asymptotic running time.

5 Experimental Results

To analyse the performance of our heuristics, we have implemented them in Java. Further, we have realized the corresponding horizontal versions to compare the resulting number of crossings with the radial algorithms. We have tested the implementations using a total number of 5000 random graphs: 50 graphs for each combination of the following parameters: $|V_1| = |V_2| \in \{20, 40, 60, 80, 100\}$ and $|E|/|V_2| \in \{1, \ldots, 20\}$.

Figure 5 shows that all radial heuristics generate fewer crossings than their horizontal equivalents, experimentally by a factor of 0.7. This is a very encouraging result, since the running times of the radial algorithms (except sifting) are similar, see Fig. 4. Like in the horizontal case [10], Cartesian barycenter on average leaves slightly fewer crossings than Cartesian median. Another similarity is that radial sifting is the best among all three radial heuristics, but also the slowest. Usually only few sifting rounds ($3-5$ for reasonable problem instances) are necessary to reach a local optimum for all vertices simultaneously, and the largest reduction of crossings usually occurs in the first round. In our experiments we further observed that the quality of radial sifting does not depend much on the quality of the initial embedding. However, a “bad” initialization raises the number of needed sifting rounds and thus the absolute running time.

Algorithm 1: RADIAL-SIFTING

Input: Two level graph $G = (V_1 \dot\cup V_2, E, \phi)$ with radial embedding $\mathcal{E} = (\pi, \psi)$
Output: Updated embedding $\mathcal{E}$, i. e., positions π_2 and offsets ψ

```
foreach $v \in V_2$ with $\deg(v) > 0$ do
    // put $v$ to the first position
    foreach $w \in V_2$ with $\pi_2(w) < \pi_2(v)$ do $\pi_2(w) \leftarrow \pi_2(w) + 1$
    $\pi_2(v) \leftarrow 0$
    let $\{v = v_0, \dots, v_{|V_2|-1}\} \leftarrow V_2$ be ordered by $\pi_2$
    let $E_v \leftarrow \{e_0, \dots, e_{\deg(v)-1}\}$ be the edges $(u, v) \in E$ ordered by $\pi_1(u)$
    // initialize offsets as 1
    foreach $e_v \in E_v$ do $\psi(e_v) \leftarrow 1$
    // initialize counters for position, offset, parting, and crossing number
    $i^* \leftarrow 0$; $j^* \leftarrow j \leftarrow 0$; $l^* \leftarrow l \leftarrow 0$; $c^* \leftarrow c \leftarrow 0$
    // search the best position for $v$
    for $i \leftarrow 0$ to $|V_2| - 2$ do
        repeat
            // try to improve the parting by reducing the offset of the next edge
            $c_1 \leftarrow \sum_{e \in E} \chi\varepsilon(e_l, e)$
            $\psi(e_l) \leftarrow j$
            $c_2 \leftarrow \sum_{e \in E} \chi\varepsilon(e_l, e)$
            // if successful, then try again, else restore the offset
            if $c_2 \leq c_1$ then
                $c \leftarrow c - c_1 + c_2$
                $l \leftarrow l + 1$
                if $l = \deg(v)$ then
                    $j \leftarrow j - 1$
                    $l \leftarrow 0$
            else $\psi(e_l) \leftarrow j + 1$
        until $c_1 < c_2$
        // remember the best position, offset, parting, and crossing number
        if $c < c^*$ then $i^* \leftarrow i$; $j^* \leftarrow j$; $l^* \leftarrow l$; $c^* \leftarrow c$
        // swap $v$ and $v_{i+1}$ and update the crossing number
        let $E_{v_{i+1}}$ be the set of edges $(\cdot, v_{i+1}) \in E$ incident to $v_{i+1}$
        $c \leftarrow c - \sum_{e_v \in E_v} \sum_{e_{v_{i+1}} \in E_{v_{i+1}}} \chi\varepsilon(e_v, e_{v_{i+1}})$
        $\pi_2(v_{i+1}) \leftarrow i$; $\pi_2(v) \leftarrow i + 1$
        $c \leftarrow c + \sum_{e_v \in E_v} \sum_{e_{v_{i+1}} \in E_{v_{i+1}}} \chi\varepsilon(e_v, e_{v_{i+1}})$
    // place $v$ to the best position
    foreach $w \in V_2$ with $\pi_2(w) \geq i^*$ do $\pi_2(w) \leftarrow \pi_2(w) + 1$
    $\pi_2(v) \leftarrow i^*$
    // set the best offsets for $v$'s incident edges
    for $i \leftarrow 0$ to $l^* - 1$ do $\psi(e_i) \leftarrow j^*$
    for $i \leftarrow l^*$ to $\deg(v) - 1$ do $\psi(e_i) \leftarrow j^* + 1$
```

6 Conclusion

We extended three well known crossing reduction techniques to radial level drawings. Further, we showed by empirical evidence, that using radial instead of horizontal level lines reduces the number of crossings significantly.

Future research on this topic can be a more efficient sifting algorithm. Also, there are some interesting problems which we do not touch in this paper: Can the number of crossings $\chi(\mathcal{E})$ in a radial embedding $\mathcal{E}$ be computed in $\mathcal{O}(\chi(\mathcal{E}))$ time? Is $\chi(\mathcal{E}) \leq 3\chi(G)$ (or similar) for an embedding $\mathcal{E}$ computed by one-sided Cartesian median heuristic on a 2-level graph G as it is for horizontal median [9]? Are there efficient radial extensions of other crossing reduction heuristics?

References

[1] C. Bachmaier, F. J. Brandenburg, and M. Forster. Radial level planarity testing and embedding in linear time. *Journal of Graph Algorithms and Applications*, 9(1):53–97, 2005.

[2] C. Bachmaier, F. Fischer, and M. Forster. Radial coordinate assignment for level graphs. In L. Wang, editor, *Proc. Computing and Combinatorics, COCOON 2005*, volume 3595 of *LNCS*, pages 401–410. Springer, 2005.

[3] M. Baur and U. Brandes. Crossing reduction in circular layout. In J. Hromkovic, M. Nagl, and B. Westfechtel, editors, *Proc. Workshop on Graph-Theoretic Concepts in Computer Science, WG 2004*, volume 3353 of *LNCS*, pages 332–343. Springer, 2005.

[4] U. Brandes, P. Kenis, and D. Wagner. Centrality in policy network drawings. In J. Kratochvíl, editor, *Proc. Graph Drawing, GD 1999*, volume 1731 of *LNCS*, pages 250–258. Springer, 1999.

[5] U. Brandes, P. Kenis, and D. Wagner. Communicating centrality in policy network drawings. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):241–253, 2003.

[6] E. G. Coffman and R. L. Graham. Optimal scheduling for two processor systems. *Acta Informatica*, 1:200–213, 1972.

[7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 2000.

[8] P. Eades and D. Kelly. Heuristics for reducing crossings in 2-layered networks. *Ars Combinatorica*, 21(A):89–98, 1986.

[9] P. Eades and N. C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(1):379–403, 1994.

[10] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25, 1997.

[11] M. Kaufmann and D. Wagner. *Drawing Graphs*, volume 2025 of *LNCS*. Springer, 2001.

[12] C. Matuszewski, R. Schönfeld, and P. Molitor. Using sifting for k-layer straightline crossing minimization. In J. Kratochvíl, editor, *Proc. Graph Drawing, GD 1999*, volume 1731 of *LNCS*, pages 217–224. Springer, 1999.

[13] E. Mäkinen. On circular layouts. *International Journal of Computer Mathematics*, 24:29–37, 1988.

[14] R. Rudell. Dynamic variable ordering for ordered binary decision diagrams. In *Proc. IEEE/ACM International Conference on Computer Aided Design, ICCAD 1993*, pages 42–47. IEEE Computer Society Press, 1993.

[15] J. M. Six and I. G. Tollis. A framework and algorithms for circular drawings of graphs. *Journal of Discrete Algorithms*, 2005. To appear, online version available.

[16] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

A Benchmark Results

The following figures provide benchmark results comparing horizontal barycenter (HB), horizontal median (HM), and horizontal sifting (HS) heuristics with their radial variants, i. e., Cartesian barycenter (CB), Cartesian median (RM), and radial sifting (RS).

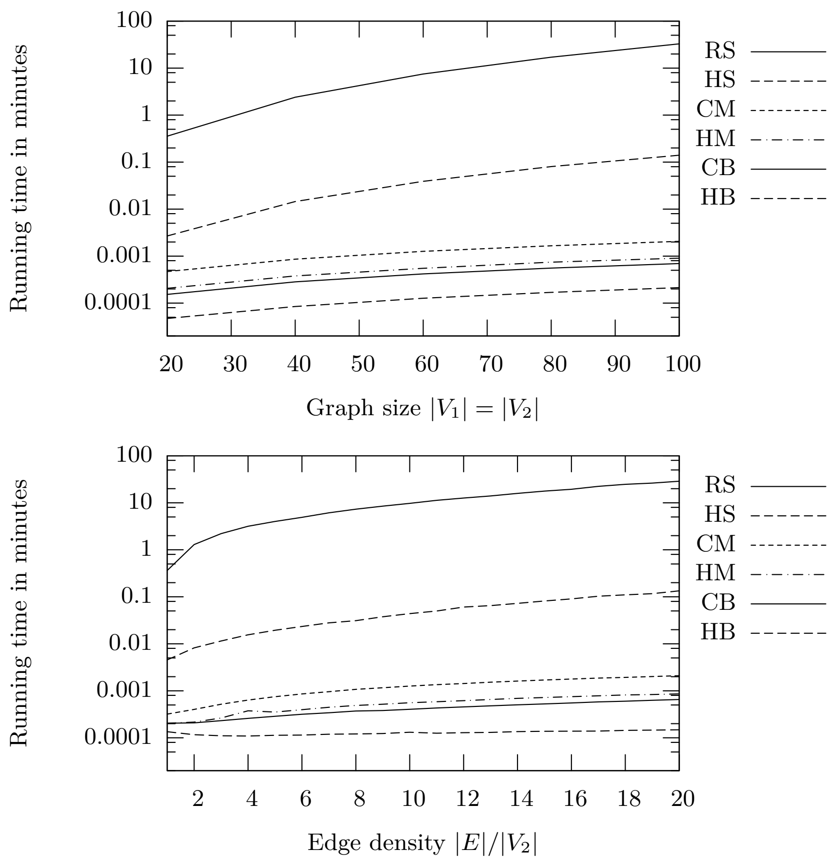


Fig. 4. Benchmark: running times

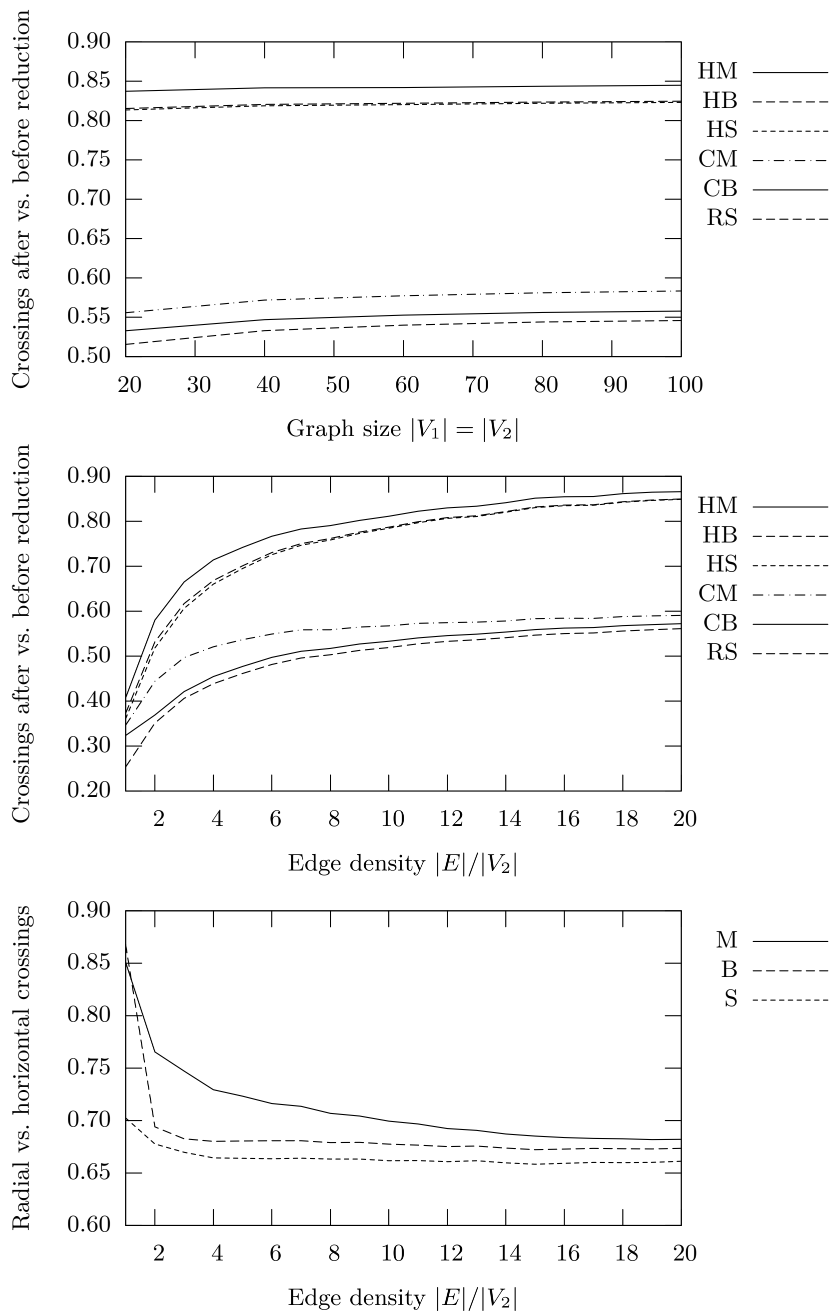


Fig. 5. Benchmark: crossing numbers