

Pinning Balloons with Perfect Angles and Optimal Area

Immanuel Halupczok and André Schulz

Institut für Mathematische Logik und Grundlagenforschung,
Universität Münster, Germany
{ihalu_01, andre.schulz}@uni-muenster.de

Abstract. We study the problem of arranging a set of n disks with prescribed radii on n rays emanating from the origin such that two neighboring rays are separated by an angle of $2\pi/n$. The center of the disks have to lie on the rays, and no two disk centers are allowed to lie on the same ray. We require that the disks have disjoint interiors, and that for every ray the segment between the origin and the boundary of its associated disk avoids the interior of the disks. Let \tilde{r} be the sum of the disk radii. We introduce a greedy strategy that constructs such a disk arrangement that can be covered with a disk centered at the origin whose radius is at most $2\tilde{r}$, which is best possible. The greedy strategy needs $O(n)$ arithmetic operations.

As an application of our result we present an algorithm for embedding unordered trees with straight lines and perfect angular resolution such that it can be covered with a disk of radius $n^{3.0367}$, while having no edge of length smaller than 1. The tree drawing algorithm is an enhancement of a recent result by Duncan et al. [Symp. of Graph Drawing, 2010] that exploits the heavy-edge tree decomposition technique to construct a drawing of the tree that can be covered with a disk of radius $2n^4$.

1 Introduction

When a graph is drawn in the plane, the vertices are usually represented as small dots. From a theoretical point of view a vertex is realized as a point, hence as an object without volume. In many applications, however, it makes sense to draw the vertices as disks with volume. The radii of the vertices can enhance the drawing by visualizing associated vertex weights [2,5]. This idea finds also applications in so-called *bubble drawings* [8], and *balloon drawings* [9,10].

Two important quality measures for aesthetically pleasant drawings are the *area* of a drawing and its *angular resolution*. The area of a drawing denotes the area of the smallest disk that covers the drawing with no edge lengths smaller than 1. The angular resolution denotes the minimum angle between two neighboring edges emanating at a vertex. Unfortunately, drawings of planar graphs with bounded angular resolution require exponential area [11]. On the other hand, by a recent result of Duncan et al. [6], it is possible to draw any unordered tree as plane straight-line graph with *perfect angular resolution*, that is the edges

incident to a vertex v are separated by an angle of at least $2\pi/\text{degree}(v)$, and polynomial area. In the same paper it was observed that an ordered tree drawn with perfect angular resolution requires exponential area. Surprisingly, even ordered trees can be drawn in polynomial area with perfect angular resolution when the edges are drawn as circular arcs [6].

The following sub-problem appears naturally in tree drawing algorithms. Suppose we have drawings of all subtrees of the children of the root. How can we group the subtrees around the root, such that the final drawing is densely packed? Often one assumes that every subtree lies exclusively in some region, say a disk. Hence, at its core, a tree drawing algorithm has to arrange disjoint disks “nicely” around a new vertex. Furthermore this task is also a fundamental base case for bubble drawing algorithms or for algorithms that realize vertices as large disks. In the paper we show how to layout the balloons with perfect angular resolution and optimal area.

More formally, let $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ be a set of n disks. To distinguish the disks B_i from other disks we call them *balloons*. The balloon B_i has radius r_i , and the balloons are sorted in increasing order of their radii. We are interested in layouts, in which the balloons of \mathcal{B} have disjoint interiors and are evenly angularly spaced. In particular, we draw for every balloon a *spoke*, that is a line segment from the origin to the balloon center. The spokes have to avoid the interior of the other balloons and two neighboring spokes are separated by an angle of $2\pi/n$. Furthermore the drawing should require only small area. We measure the area of the balloon layout by the radius of the smallest disk that is centered at the origin and covers all balloons.

Results. We show how to locate the balloons with perfect angular resolution such that the drawing can be covered with a disk of radius $2\tilde{r}$, for \tilde{r} being the sum of the radii. This is clearly the best possible result in the worst case, since when $|\mathcal{B}| = 1$, the area of the best balloon layout is clearly $2r_1$. We also study a modified version of the balloon layout problem that finds application in a tree drawing algorithm. Here, one and two spokes may remain without balloon, but the angle between the two unused spokes has to be at least $2\pi/3$. In this setting we obtain a balloon drawing that can be covered with a disk of radius $(1 + \sqrt{2 - 2/\sqrt{5}})\tilde{r} \approx 2.0514\tilde{r}$. The induced algorithm draws unordered trees with perfect angular resolution and with area smaller than $n^{3.0367}$.

Related work. Without explicitly stated, Duncan et al. [6] studied the balloon layout problem (with one or two unused spokes) as part of their drawing algorithm for unordered trees and obtained a bound of $4\tilde{r}$ for the area. The induced tree drawing algorithm produces drawings with area smaller than $2n^4$. For the special case of orthogonal straight-line drawings of ternary trees (they automatically guarantee perfect angular resolution) Frati [7] provided an algorithm whose drawings require $O(n^{1.6131})$ area; the drawing of the complete ternary tree requires $O(n^{1.262})$ area. Bachmaier et al. obtained a drawing of the complete 6-regular tree with perfect angular resolution with area $O(n^{1.37})$ [1].

In contrast to our setting the so-called balloon drawings [9,10] place all balloons at the same distance. Also related are the (non-planar) *ringed circular layouts* [13]. Without the perfect angular resolution constraint trees can be drawn with area $O(n \log n)$ [4].

Conventions. We normalize the radii of the balloons such that they sum up to 1. In intermediate stages of the drawing algorithm a spoke may be without a balloon. In this case we consider the spoke as a ray emanating from the origin that fulfills the angular resolution constraint. When we say that “we place balloon B on s at distance x ” we mean that the balloon B is placed on a spoke s (that had no associated balloon yet) such that its center lies on s at Euclidean distance x from the origin. In the remainder of the paper all disks covering the balloons are considered as centered at the origin.

2 The Greedy Strategy

In the following section we introduce the greedy strategy for placing \mathcal{B} with perfect angles. To keep things simple we assume for now that the number of balloons n is a power of two. The general case is discussed later.

We place the balloons in increasing order of their radii. Thus we start with the smallest balloon and end with the largest balloon. The placement of the balloons is carried out in *rounds*. In every round we locate half of the balloons that have not been placed yet. Thus, we “consume” a certain number of spokes in each round. Let S be the list of spokes that are available in the beginning of a round in cyclic order. In every round we select every other spoke as a spoke on which a balloon is placed in the current round. This ensures that consecutive spokes that receive a balloon in round i are separated by an angle of $\alpha_i := 2^{i+1}\pi/n$. For every round we define the *safe disk* SD_i centered at the origin with radius safe_i . The safe disk is the smallest disk covering all balloons that were placed in previous rounds. In round i we place all balloons such that they avoid the interior of the safe disk SD_i . Thus, the best we can hope for is to place the balloons such that they touch SD_i . Whenever this is possible we speak of a *contact situation*, depicted in Figure 1(a). The safe disks ensure that balloons placed in the current round will not intersect the interior of the balloons that were placed in previous rounds. However, we have to guarantee that balloons placed in the same round will also not interfere with the remaining spokes. Suppose that B_j is assigned to the spoke s_k . We enforce B_j to lie inside a wedge with opening angle α_i centered at s_k . This wedge is named W_k . Since the spokes that are used in round i are separated by α_i , the wedges of round i have disjoint interiors. Whenever a balloon touches the boundary of its associated wedge we speak of a *wedge situation*, as shown in Figure 1(b).

The greedy strategy tries first to place B_j at its spoke s_k , such that it touches SD_i . If this would imply that B_j is not contained inside W_k , we move the center of B_j on s_k away from the origin, until B_j touches the boundary of W_k . In case a wedge situation occurs, we can compute the location of the center of B_j with help of the following lemma, whose proof can be found in Duncan et al. [6].

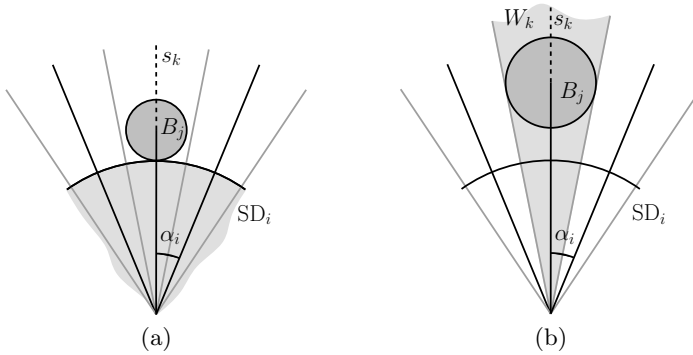


Fig. 1. In a contact situation (a) we place B_i such that it touches SD_i . In contrast, in a wedge situation (b), we place B_i such that it touches the boundary of W_k (when it is placed on s_k).

Lemma 1. *Let W be a wedge with opening angle φ centered at a spoke s . Further let B be a balloon with radius r that is placed such that (1) its center lies on s , and (2) it touches the boundary of W . Then B is contained inside a disk centered at the origin with radius*

$$(1 + \sin(\varphi/2))/(\sin(\varphi/2)) \cdot r.$$

In the remainder of the paper we use as notation

$$\alpha(\varphi) := (1 + \sin(\varphi/2))/(\sin(\varphi/2)). \tag{1}$$

Notice that when a wedge situation occurs in round i , then in particular a wedge situation has to occur for the last balloon that is added in round i , since the balloons are sorted by increasing radii. All balloons placed in round i are sandwiched between SD_i and SD_{i+1} . We call the region $SD_{i+1} \setminus SD_i$ the i -th layer L_i .¹ The *width* of layer L_i is defined as $\text{safe}_{i+1} - \text{safe}_i$. When a wedge situation occurs in round i , the layer L_i is called a *wedge layer*, otherwise a *contact layer*. An example of a wedge layer is shown in Figure 2.

2.1 Splitting the Set of Spokes

We come now back to the case where n is not necessarily a power of two. In this setting there might be an odd number of spokes in some round. In such a round we place only $\lfloor k/2 \rfloor$ balloons, such that no two of them are assigned to consecutive spokes. This however has two drawbacks: First, the angles might not split evenly, and second, the layers will be filled with less balloons.

We can always pick $\lfloor k/2 \rfloor$ spokes such that in the remaining set of spokes at most two separating angles are smaller than the others, which are all equal.

¹ By convention $SD_1 = \emptyset$, and for i being the last round, $SD_{i+1} =$ smallest disk covering all balloons.

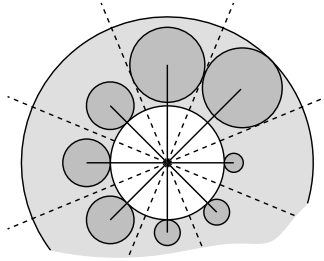


Fig. 2. A wedge layer (shaded) that had been filled with balloons by the greedy strategy

Moreover, the two smaller angles are each at least half as big as the remaining angles. We call every set of spokes for which this property holds *well-separated*. Furthermore we assume that a well-separated set of spokes is ordered such that the two smaller angles are realized between the first and second, and between the second and third spoke. Algorithm 1 describes a strategy that picks $\lfloor k/2 \rfloor$ of the spokes and ensures that the remaining set of spokes is still well-separated if the original set was well-separated.

Algorithm 1. SplitSpokes(S)

Input : S set of spokes

Output: (T, T') , such that T' are the spokes that will be used in the current round, $T = S \setminus T'$.

- 1 $T' \leftarrow$ every spoke of S with even index
 - 2 $T \leftarrow S \setminus T'$
 - 3 reorder T by putting the last spoke in front
 - 4 **return** (T, T')
-

Lemma 2. *Let S be a well-separated set of at least three spokes and let φ denote the size of the big angles in S . Let (T, T') be the return value of Algorithm 1.*

- (1) *If $|T| > 2$, then T is well-separated.*
- (2) *If $|T| = 2$, then the smaller angle between the two spokes is at least $2\pi/3$.*
- (3) *The wedge with angle φ centered at the first spoke in T' contains no spoke of S in its interior.*
- (4) *A wedge with angle 2φ centered at a spoke in T' that is not the first spoke contains no spoke of S in its interior.*

Proof. Let the angle between the first and second spoke in S be γ_1 , and let the angle between the second and third spoke in S be γ_2 . Since S is well-separated, we have $\varphi/2 \leq \gamma_1, \gamma_2 \leq \varphi$. Hence the wedge centered at the second spoke of S with angle φ does not contain any other spoke of S in its interior, which proves (3). Property (4) is due to the fact that every spoke in S with even index larger than 2 is separated from its neighboring spokes by an angle of φ .

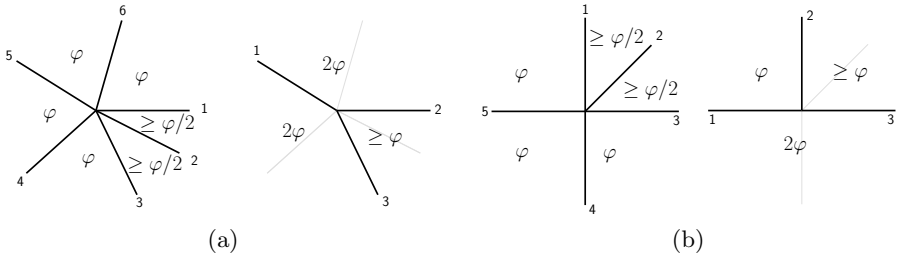


Fig. 3. Merging the angles as implied by Algorithm 1. In case we have an even number of spokes (a), and in case we have an odd number of spokes (b). The spoke numbers are shown as small numbers.

After line 2 of Algorithm 1, the angle between the first and second spoke of T equals $\gamma_1 + \gamma_2 \geq \varphi$. In case S has an even number of spokes the remaining angles of size φ are grouped pairwise and therefore the corresponding angles in T are all 2φ , which proves property (1) for this case. If the set S contains an odd number of spokes, the additional angle between the last spoke in T and the first spoke in T is also φ . Hence after reordering, the new set T is again well-separated, and (1) follows. Figure 3 illustrates the outcome of Algorithm 1.

To see that (2) is true, notice the following. T contains two spokes, if S contains three or four spokes. In case S contains 4 spokes, the sum of the two small angles is at least $2\pi/3$. In case S contains three spokes, the sum of the two small angles between the spokes is at least π . The large angle between the spokes in S is at least $2\pi/3$. This angle appears also between the spokes in T . \square

To ensure that the balloons of each layer cannot interfere with each other and with the remaining spokes, we place them inside the wedges defined by Lemma 2(3–4). All wedges have the same opening angle, say φ , except the first wedge, whose opening angle is at least $\varphi/2$. The balloon with the smallest radius in each round is placed inside the wedge with the (possible) smaller opening angle.

2.2 The Final Layer

It is important to analyze the situation where the greedy strategy has to stop. In every round we reduce the number of spokes from k to $\lceil k/2 \rceil$. If we subdivide the spokes in this fashion we will come to a point where exactly two spokes are left. The final two balloons are placed in the last round as follows: (1) The balloon B_n will be placed such that it touches the safe disk. (2) The balloon B_{n-1} will be placed such that it is contained inside a wedge with opening angle $\pi/3$, centered at its spoke, while avoiding the interior of the current safe disk.

Lemma 3. *When the balloons are placed as discussed in the previous paragraph, then one of the following is true:*

1. *The width of the last layer is $2r_n$.*
2. *All balloons can be covered with a disk of radius two.*

Proof. Let φ be the smaller of the two angles between the spokes in the final round i . Due to Lemma 2, φ is at least $2\pi/3$. The tangent of B_n at its intersection with SD_i separates B_n from the spoke of B_{n-1} . Since the angle between this tangent and the spoke of B_{n-1} is at least $\varphi - \pi/2 \geq \pi/6$ it is safe to place B_{n-1} inside a wedge centered at its spoke with opening angle $\pi/3$. Thus, either B_{n-1} touches SD_i , or it is contained inside a disk of radius $\alpha(\pi/3)r_{n-1} = 3r_{n-1}$. In the former case the width of the layer is $2r_n$, in latter case the radius of the covering disk is at most $\max\{2r_n, 3/2\}$ (recall that $r_{n-1} \leq 1/2$). \square

Due to Lemma 3 we can assume that the width of the last layer equals $2r_n$. Thus even if B_{n-1} defines a wedge situation we consider the last layer as contact layer. We summarize the discussion in Algorithm 2.

Algorithm 2. GreedyBalloon(S)

```

Input :  $S$ : spokes in cyclic order.
1  $k \leftarrow 0$  // number of balloons placed so far
2  $\text{safe} \leftarrow 0$  // radius of the current safe disk
3 while  $|S| > 2$  do
4    $(T, T') \leftarrow \text{SplitSpokes}(S)$ 
5    $\text{width} \leftarrow 0$  // width of the current layer so far
6   for  $i \leftarrow k + 1$  to  $k + |T'|$  do
7      $s \leftarrow (i - k)$ -th spoke of  $T'$ 
8      $\varphi \leftarrow 2(\text{minimal angle between } s \text{ and one of its neighboring spokes in } S)$ 
9      $c \leftarrow \max\{\alpha(\varphi)r_i - r_i, \text{safe} + r_i\}$  // center of  $B_i$ 
10    place  $B_i$  on  $s$  at distance  $c$ 
11     $\text{width} \leftarrow \max\{\text{width}, c + r_i - \text{safe}\}$ 
12  end
13   $\text{safe} \leftarrow \text{safe} + \text{width}$ 
14   $k \leftarrow k + |T'|$ 
15   $S \leftarrow T$ 
16 end
17 let  $s_1, s_2$  be the spokes in  $S$ 
18 place  $B_n$  on  $s_1$  at distance  $\text{safe} + r_n$ 
19 place  $B_{n-1}$  on  $s_2$  at distance  $\max\{2r_{n-1}, \text{safe} + r_{n-1}\}$ 

```

2.3 Quality of the Greedy Strategy

We denote by R the radius of the smallest disk that covers all balloons. In order to determine R we have to consider only certain radii.

Lemma 4. *The radius of the smallest disk R that covers all balloons drawn with Algorithm 2 can be determined with the knowledge of*

1. the number of spokes,
2. the radius of the largest and smallest balloon in the outermost wedge layer,
3. the radii of the largest balloons in each of the contact layers following the outermost wedge layer.

Proof. Suppose the last wedge situation occurs in round i . Then the radius of SD_{i+1} is determined by a balloon that touches its wedge. All wedges have the same opening angle, except maybe the first wedge. Since the smallest balloon is placed inside the first wedge, the wedge situation that defines the radius of SD_{i+1} depends on the possible wedge situation of the largest and smallest balloon only. The following layers are all contact layers. Their width is determined by the diameter of the largest balloons in each layer. The radius R equals therefore the radius of SD_{i+1} with the addition of the widths of the following contact layers. \square

Since we are interested in a worst case bound for R we make the following assumptions to simplify the analysis of the algorithm.

Lemma 5. *Let r_w be the radius of the balloon, whose wedge situation determined the width of the last wedge layer L_k . The radius R of the smallest covering disk is maximized when*

$$r_w = r_{w+1} = r_{w+2} = \dots = r_{n-1}, \text{ and}$$

$$r_1 = r_2 = r_3 = \dots = r_{w-1} = 0.$$

Proof. We consider the radii as resources that we want to spend to make R as large as possible. Since no radius of a balloon with smaller index than w matters for R , we set these radii to zero to save resources. If B_w is the smallest balloon in its layer, all radii of balloons in L_k have the same radius in the worst case. Otherwise we could shrink some of these balloons without changing the width of L_k and spent the resources to increase r_n and therefore R .

Only the balloon added last in each contact layer determines the width of its layer. We select the radii of the other balloons in contact layers as small as possible, i.e., as large as the radius of the largest balloon in the previous layer. If any of these radii would be larger we could make such a radius smaller and increase r_n instead, which would increase R .

Assume we have at least two contact layers following L_k . Let B_c be the largest balloon in the contact layer L_{k+1} , that is the balloon last added in L_{k+1} . Due to the discussion in the previous paragraph we can assume that the balloon B_{c+1} in the next layer has radius r_c . If $r_c > r_w$, we could lower the radius by $r_c - r_w$ for B_c and B_{c+1} each. By this we can increase r_n by $2(r_c - r_w)$. As a consequence the radius R increases by $r_c - r_w$. Therefore in the worst case all radii in layer L_{k+1} equal r_w . By an inductive argument the radii in the last contact layers are all r_w . The only exception is the largest balloon B_n . \square

Theorem 1. *Algorithm 2 constructs a drawing of balloons with disjoint interiors and spokes that intersect only the interior of their associated balloon that can be covered with a disk of radius two, which is best possible.*

Proof. We define as \bar{L}_i the i -th last layer such that \bar{L}_1 is the last layer. Suppose there were ℓ spokes left, before the last wedge layer was filled. We denote the

number of contact layers that follow the last wedge layer by k . By Algorithm 1 the number k is given by a function $k = f(\ell)$, which is defined as follows

$$f(\ell) := \begin{cases} 1 & \text{if } 3 \leq \ell \leq 4, \\ 1 + f\left(\frac{\ell}{2}\right) & \text{if } \ell > 4, \text{ even,} \\ 1 + f\left(\frac{\ell+1}{2}\right) & \text{if } \ell > 4, \text{ odd.} \end{cases} \tag{2}$$

By induction, $f(\ell) \leq \log(\ell - 1)$. The radius of the covering disk R equals the radius of \bar{L}_k 's safe disk plus the width of the last k contact layers. Let B_w be the balloon that determined safe_k . By Lemma 5 we can assume that all balloons following B_w have radius r_w , except B_n . All other radii are zero.

As previously discussed, the balloon B_w is either the first or the last balloon in the last wedge layer. We discuss the two possibilities by case distinction. Let us first assume that B_w is the last balloon of layer \bar{L}_{k+1} . By construction the last balloon is placed inside the wedge with largest opening angle (in this round). Therefore its opening angle φ is minimized, when the angles between all pairs of neighboring spokes are equal. We have ℓ spokes in \bar{L}_{k+1} , and therefore two spokes are separated by $2\pi/\ell$ and $\varphi = 4\pi/\ell$. Furthermore, we have $k - 1$ layers of width $2r_w$, and one layer of width $2r_n$ following \bar{L}_{k+1} . In layer \bar{L}_{k+1} we place no more than $\ell/2$ balloons and therefore in the last k layers we have at least $\ell/2$ balloons in total. Since there is one balloon in \bar{L}_{k+1} with radius r_w and only one balloon in the last k layers with radius different from r_w , we get $r_n \leq 1 - r_w\ell/2$. This leads to

$$R \leq \alpha(\varphi)r_w + 2(k - 1)r_w + 2r_n \leq 2 + [\alpha(4\pi/\ell) + 2 \log(\ell - 1) - \ell - 2] r_w.$$

The last wedge layer must contain at least three spokes. Since $\alpha(4\pi/\ell) + 2 \log(\ell - 1) - \ell - 2$ is decreasing² for $\ell \geq 4$ and negative for $\ell = 3, 4$, we get $R \leq 2$.

We assume now that B_w was placed first in \bar{L}_{k+1} . Again, let φ be the angle of the wedge that contains B_w centered at its spoke. Due to Lemma 2 the angles between two neighboring spokes are all of size ψ except two angles, which are at least $\psi/2$ (the small angles). The angle φ is twice the minimum of the two small angles, and hence minimized when one of the small angles has size ψ and the other has size $\psi/2$. In this case we have $\ell - 1$ angles of size ψ and one angle of size $\psi/2$. Since all angles sum up to 2π , we have $\psi = 2\pi/(\ell - 1/2)$, which is a lower bound for φ . Notice that all balloons in \bar{L}_{k+1} have now radius r_w , hence we have $\ell - 1$ balloons of radius r_w , and therefore $r_n \leq 1 - (\ell - 1)r_w$. We conclude with

$$R \leq \alpha(\varphi)r_w + 2(k - 1)r_w + 2r_n \leq 2 + [\alpha(2\pi/(\ell - 1/2)) + 2 \log(\ell - 1) - 2\ell] r_w.$$

For $\ell \geq 2$ the expression $\alpha(2\pi/(\ell - 1/2)) + 2 \log(\ell - 1) - 2\ell$ is negative and decreasing and the theorem follows. \square

² The estimation of this expression and of similar following expressions was obtained by computer algebra software.

3 Drawing Unordered Trees with Perfect Angles

The greedy strategy can be used to construct drawings of unordered trees with perfect angular resolution and small area. In fact, the balloon layout problem studied in Section 2 is a subproblem of the drawing algorithm of Duncan et al. [6], where it is used to draw depth-1 trees. With the help of the so called heavy edge tree-decomposition (see Tarjan [12]) these trees are combined to the original tree. Since our proposed strategy uses significantly smaller area, it implies an improvement for the area of the tree drawing.

We start with a brief review of the heavy edge tree-decomposition. Let u be a non-leaf of the rooted tree T . We denote by T_u the subtree of T rooted at u . Let v be the child of u such that T_v has the largest number of nodes (compared to the subtrees of the other children of u), breaking ties arbitrarily. We call the edge (u, v) a *heavy edge*, and the edges to the other children of u *light edges*. The heavy edges induce a decomposition of T into paths, called *heavy paths*, and light edges; see Figure 4 on the left.

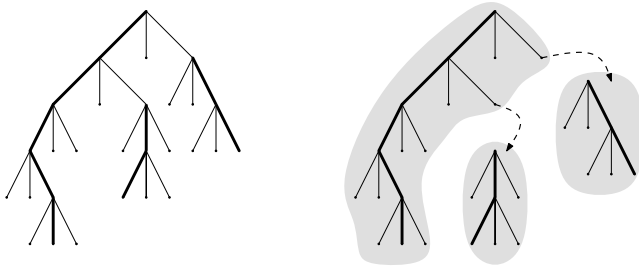


Fig. 4. An example of a heavy-edge tree-decomposition

In order to draw a tree with perfect angular resolution we have to modify the greedy strategy such that (1) one or two spokes can remain without balloon, and (2) the separating angle between the two unused spokes is at least $2\pi/3$. This however, comes at a cost, we have to make the disk that covers the drawing slightly larger. Thus instead of a disk with radius 2 we might need a disk of radius κ , where $\kappa = (1 + \sqrt{2 - 2/\sqrt{5}}) \approx 2.0514$. Notice that we use an additional construction, which can be found in the full paper, to make the balloon packing slightly denser.

By construction, every non-leaf tree node lies on exactly one heavy path. Let C be the union of the heavy path that is incident to the root with its incident light edges. By deleting C the original tree splits into subtrees. Assume that we have constructed the drawings for these subtrees by recursion. We are left with drawing C such that the leaves of C are drawn as disjoint disks. The disk radii are chosen such that each drawing of a subtree fits inside its associated disk. For every node on the heavy path we apply the greedy strategy to draw the associated disks as balloons while leaving the heavy edges as free spokes without a disk. These drawings of depth-1 trees can be combined by the strategy of

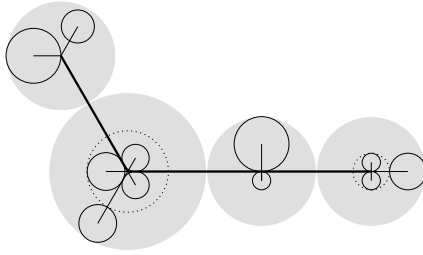


Fig. 5. Drawing of the root-heavy path with incident light edges and *safe regions* for the missing subtrees

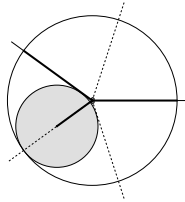


Fig. 6. Three balloons with radius $\varepsilon, \varepsilon, 1-2\varepsilon$ and 5 spokes. Separating the unused spokes by an angle $\geq 2\pi/3$ yields a covering disk with radius $\alpha(2\pi/5) = \kappa$. when $\varepsilon \rightarrow 0$.

Duncan et al.[6, full version, Lemma 2.3]. Figure 5 illustrates this construction. Notice that for every layer of the recursion we have to scale the intermediate drawing by a factor of 2κ . However, since by construction every root-leaf path in the original tree visits at most $\log n$ light edges, the recursion depth is logarithmic. A more detailed analysis, which can be found in the full version of the paper, proves the following theorem.

Theorem 2. *Using Algorithm 2 in the framework of Duncan et al. produces a drawing of an unordered tree with n nodes that has perfect angular resolution and that can be covered with a disk of radius $n^2 \cdot n^{\log \kappa} < n^{3.0367}$, while having no edge with length smaller 1.*

4 Concluding Remarks

The algorithm presented in this paper runs in linear time. Notice that even when the set \mathcal{B} is not ordered by radii we can obtain a running time of $O(n)$. In fact, Algorithm 2 works correctly when \mathcal{B} is *weakly ordered* by radii, that is, (1) the median and the smallest element are in the “right” position, (2) every radii between median and smallest element is not larger than the median, and (3) the sequence of elements larger than the median are weakly ordered. Since the median and the smallest radius can be found in $O(n)$ time [3], the recursive definition implies that \mathcal{B} can be weakly ordered in $O(n)$ time.

The only case, where we obtain strict inequalities in the proof of Theorem 1, is when $|\mathcal{B}| = 1$. By placing all balloons slightly inside the wedges, resp., slightly outside the safe disks we can therefore modify all constructions such that no balloons touch.

As a final remark we point out that Theorem 1 can be generalized such that it holds for one or two unused spokes, while guaranteeing that the whole balloon drawing can be covered with a disk of radius 2. However, as depicted in Figure 6, the slightly worse bound of κ cannot be avoided if one has to guarantee that the smaller angle between the two unused spokes is at least $2\pi/3$.

References

1. Bachmaier, C., Brandenburg, F.-J., Brunner, W., Hofmeier, A., Matzeder, M., Unfried, T.: Tree Drawings on the Hexagonal Grid. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 372–383. Springer, Heidelberg (2009)
2. Barequet, G., Goodrich, M.T., Riley, C.: Drawing Graphs with Large Vertices and Thick Edges. In: Dehne, F.K.H.A., Sack, J.-R., Smid, M.H.M. (eds.) WADS 2003. LNCS, vol. 2748, pp. 281–293. Springer, Heidelberg (2003)
3. Blum, M., Floyd, R.W., Pratt, V.R., Rivest, R.L., Tarjan, R.E.: Time bounds for selection. *J. Comput. Syst. Sci.* 7(4), 448–461 (1973)
4. Crescenzi, P., Battista, G.D., Piperno, A.: A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry: Theory & Application Geom.* 2, 187–200 (1992)
5. Duncan, C.A., Efrat, A., Kobourov, S.G., Wenk, C.: Drawing with fat edges. *Int. J. Found. Comput. Sci.* 17(5), 1143–1164 (2006)
6. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Nöllenburg, M.: Drawing Trees with Perfect Angular Resolution and Polynomial Area. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, vol. 6502, pp. 183–194. Springer, Heidelberg (2011), <http://arxiv.org/pdf/1009.0581v1>
7. Frati, F.: Straight-Line Orthogonal Drawings of Binary and Ternary Trees. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) GD 2007. LNCS, vol. 4875, pp. 76–87. Springer, Heidelberg (2008)
8. Grivet, S., Auber, D., Domenger, J.P., Melancon, G.: Bubble tree drawing algorithm. In: International Conference on Computer Vision and Graphics, pp. 633–641. Springer, Heidelberg (2004)
9. Lin, C.-C., Yen, H.-C.: On balloon drawings of rooted trees. *Journal of Graph Algorithms and Applications* 11(2), 431–452 (2007)
10. Lin, C.-C., Yen, H.-C., Poon, S.-H., Fan, J.-H.: Complexity analysis of balloon drawing for rooted trees. *Theor. Comput. Sci.* 412(4-5), 430–447 (2011)
11. Malitz, S.M., Papakostas, A.: On the angular resolution of planar graphs. *SIAM J. Discrete Math.* 7(2), 172–183 (1994)
12. Tarjan, R.E.: Linking and cutting trees. In: Data Structures and Network Algorithms, ch. 5, pp. 59–70. SIAM (1983)
13. Teoh, S.T., Ma, K.-L.: RINGS: A Technique for Visualizing Large Hierarchies. In: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 268–275. Springer, Heidelberg (2002)