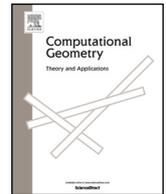


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# Computational Geometry: Theory and Applications

[www.elsevier.com/locate/comgeo](http://www.elsevier.com/locate/comgeo)


## On the recognition and reconstruction of weighted Voronoi diagrams and bisector graphs



Günther Eder, Martin Held\*, Stefan de Lorenzo, Peter Palfrader

Universität Salzburg, FB Informatik, Salzburg, Austria

### ARTICLE INFO

#### Article history:

Received 18 February 2021

Received in revised form 10 August 2022

Accepted 17 August 2022

Available online 24 August 2022

#### Keywords:

Voronoi diagram

Bisector graph

Recognition

Reconstruction

Multiplicative weight

### ABSTRACT

A weighted bisector graph is a geometric graph whose faces are bounded by edges that are portions of multiplicatively weighted bisectors of pairs of (point) sites such that each of its faces is defined by exactly one site. A prominent example of a bisector graph is the multiplicatively weighted Voronoi diagram of a finite set of points which induces a tessellation of the plane into Voronoi faces bounded by circular arcs and straight-line segments. Several algorithms for computing various types of bisector graphs are known. In this paper we reverse the problem: Given a partition  $\mathcal{G}$  of the plane into faces, find a set of points and suitable weights such that  $\mathcal{G}$  is a bisector graph of the weighted points, if a solution exists. If  $\mathcal{G}$  is a graph that is regular of degree three then we can decide in  $\mathcal{O}(m)$  time whether it is a bisector graph, where  $m$  denotes the combinatorial complexity of  $\mathcal{G}$ . In the same time we can identify up to two candidate solutions such that  $\mathcal{G}$  could be their multiplicatively weighted Voronoi diagram. Additionally, we show that it is possible to recognize  $\mathcal{G}$  as a multiplicatively weighted Voronoi diagram and find all possible solutions in  $\mathcal{O}(m \log m)$  time if  $\mathcal{G}$  is given by a set of disconnected lines and circles.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

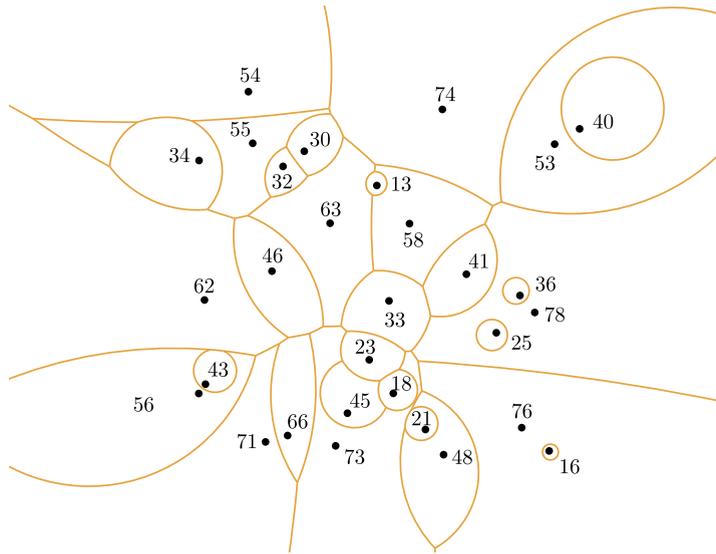
### 1.1. Motivation and related work

A *geometric graph* is the fixed embedding of a planar graph in the plane so that its nodes are represented by points and all its edges belong to some specific family of curves, e.g., straight-line segments and circular arcs. Following [1], we refer to a geometric graph as a *bisector graph* if there exists a set of input sites such that all edges of the graph lie on bisectors of pairs of sites. Furthermore, it is required that every face of a bisector graph is defined by exactly one site. (See the end of Section 1.2 for a definition of this concept.) Several authors deal with the computation of particular types of weighted bisector graphs and present strategies to construct them efficiently [2–4].

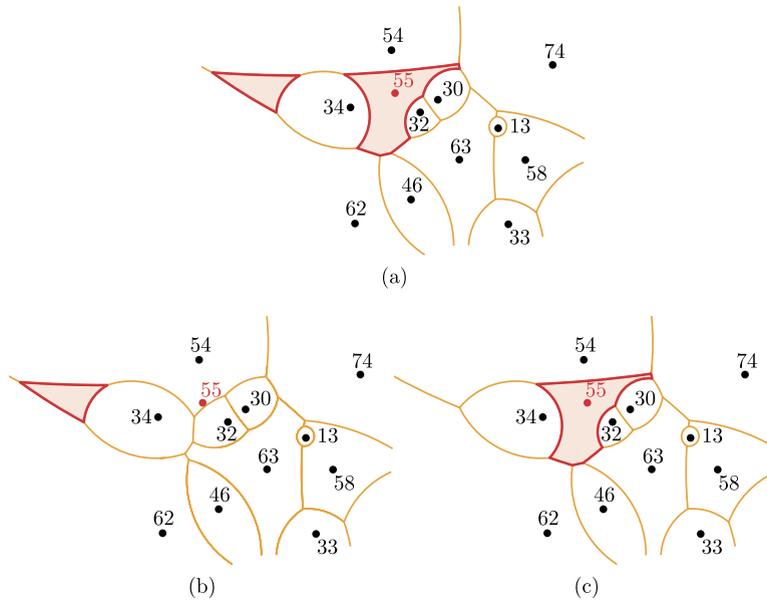
In this work we focus on bisector graphs that correspond to a set  $S$  of weighted points in the plane and study the reverse problem: Given  $\mathcal{G}$ , a geometric graph that allegedly is a weighted bisector graph, can we recognize  $\mathcal{G}$  as such, and if so, can we reconstruct the respective input sites  $S$  and weights  $\sigma$  such that the resulting bisector graph is equal to  $\mathcal{G}$ ? Furthermore, we will discuss several settings in which we are even able to recognize  $\mathcal{G}$  as a special type of bisector graph that is known as *multiplicatively weighted Voronoi diagram*; see Figs. 1 and 2.

\* Corresponding author.

E-mail addresses: [geder@cs.sbg.ac.at](mailto:geder@cs.sbg.ac.at) (G. Eder), [held@cs.sbg.ac.at](mailto:held@cs.sbg.ac.at) (M. Held), [slorenzo@cs.sbg.ac.at](mailto:slorenzo@cs.sbg.ac.at) (S. de Lorenzo), [palfrader@cs.sbg.ac.at](mailto:palfrader@cs.sbg.ac.at) (P. Palfrader).



**Fig. 1.** The multiplicatively weighted Voronoi diagram (in orange) of 30 input sites in which the point locations are highlighted by the black dots. The corresponding weights are written next to them. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)



**Fig. 2.** In (a) a section of the multiplicatively weighted Voronoi diagram that is depicted in Fig. 1 is shown. The region of the site  $s$  that is associated with weight 55 (highlighted in red) consists of two connected components. Furthermore, (b) and (c) show two different bisector graphs in which  $s$  is only associated with exactly one connected component.

Multiplicatively weighted Voronoi diagrams of points in the plane were first introduced by Boots [5]. Aurenhammer and Edelsbrunner [2] present a worst-case optimal algorithm to compute the multiplicatively weighted Voronoi diagram under the Euclidean distance. Algorithms with a decent expected-case complexity are due to Har-Peled and Raichel [3] and Held and de Lorenzo [4]. Har-Peled and Raichel [3] also prove that the expected combinatorial complexity of the multiplicatively weighted Voronoi diagram is bounded by  $\mathcal{O}(n \log^2 n)$  if the weights of all input points are chosen randomly. Eder and Held [6] describe an incremental algorithm for constructing the multiplicatively weighted Voronoi diagram under the maximum norm.

Multiplicatively weighted Voronoi diagrams are widely used in wireless communication to model the coverage areas of sensors or transmitters. If the devices are heterogeneous and distance to a device is measured by means of the Euclidean distance weighted by the sensing/transmitting power of the device then the service areas can be modeled as the regions of (multiplicatively weighted) Voronoi diagrams of the device positions. See, e.g., [7–9].

The problem studied is motivated by a problem forwarded to us by a company working on wireless sensor networks: They get a geometric graph  $\mathcal{G}$ , a set of sensor positions  $S$  and weights  $\sigma$  from an application of one of their customers. The data received is of a low quality, with very low precision of all numerical values, such that a subsequent analysis reveals inconsistencies. That is, the graph  $\mathcal{G}$  and the weighted Voronoi diagram of  $S$  do not seem to match. Taking only  $S$  and  $\sigma$  as input and (re-)computing the corresponding Voronoi diagram is no option since it may be strikingly different from  $\mathcal{G}$ . This is no surprise because it is known that minor changes in the positions or weights of point sites may change their Voronoi diagram substantially. Hence, the company's next-best idea was to take  $\mathcal{G}$  and try to reconstruct  $S$  and  $\sigma$ .

Ash and Bolker [10] were among the first to study the recognition problem for unweighted Voronoi diagrams of point sites. Harvingsten [11] presents a polynomial-time algorithm that is based on linear programming, for recognizing whether a given tessellation of  $\mathbb{R}^d$  is an unweighted Voronoi diagram, and reconstructing the respective set of  $d$ -dimensional input points. Aurenhammer's work [12] on reciprocal figures and projection polyhedra also allows to characterize and recognize Voronoi diagrams in higher dimensions. Biedl et al. [13] present a strategy for reconstructing the polygon or planar straight-line graph from a given straight-skeleton or Voronoi diagram in  $\mathcal{O}(n \log n)$  time, where  $n$  is the number of edges of the input graph. Aichholzer et al. [14,15] investigate the realizability of a tree as the straight skeleton of a polygon. Eder et al. [16] explain how to reconstruct weighted straight skeletons from geometric trees.

### 1.2. Preliminaries

Let  $S$  be a finite set of  $n$  distinct point sites and denote their weight function by  $\sigma : S \rightarrow \mathbb{R}^+$ . That is,  $\sigma(s)$  specifies the weight of the site  $s \in S$ . For a point  $p$  in the Euclidean plane and a site  $s \in S$ , the (weighted) distance from  $p$  to  $s$  is defined as

$$d_\sigma(p, s) := \frac{d(p, s)}{\sigma(s)},$$

where  $d(\dots)$  denotes the standard Euclidean distance. Of course,  $d_\sigma(p, S) := \min\{d_\sigma(p, s) : s \in S\}$ . We follow common Voronoi terminology and define the (weighted) Voronoi region of  $s \in S$  as the set of all points in  $\mathbb{R}^2$  that are not farther from  $s$  than from any other site of  $S$  with respect to  $d_\sigma$ :

$$\mathcal{R}_\sigma(s, S) := \{p \in \mathbb{R}^2 : d_\sigma(p, s) \leq d_\sigma(p, S)\}.$$

Then the weighted Voronoi diagram  $\mathcal{VD}_\sigma(S)$  of  $S$  relative to  $\sigma$  is the union of all region boundaries. In the sequel, we will simplify the terminology by dropping the term "weighted" from a Voronoi diagram and related structures and, e.g., simply refer to  $\mathcal{VD}_\sigma(S)$  as Voronoi diagram of  $S$ . Note that  $\mathcal{VD}_\sigma(S)$  may have a quadratic combinatorial complexity [2].

Consider an embedding of a planar geometric graph  $\mathcal{G}$  in  $\mathbb{R}^2$  such that its edges are formed by circular arcs, full circles and straight lines. All end-points of the circular arcs form the *nodes* of  $\mathcal{G}$ . We demand that all edges of  $\mathcal{G}$  meet only at these nodes and that each node has a degree at least three. We call such an embedding a *planar circular-arc graph*. The number of faces of the planar subdivision induced by  $\mathcal{G}$  is denoted by  $m$ . Euler's Theorem for planar graphs implies that  $\mathcal{G}$  has  $\mathcal{O}(m)$  nodes and  $\mathcal{O}(m)$  edges. In the sequel we will use  $\mathcal{G}$  as our input that we seek to recognize.

As usual, a weighted bisector between two sites  $s_1, s_2 \in S$  is the locus of points that have the same weighted distance from  $s_1$  and  $s_2$ . A (*weighted*) *bisector graph* of  $(S, \sigma)$  is a planar circular-arc graph  $\mathcal{G}$  such that (1) all edges of  $\mathcal{G}$  lie on weighted bisectors of  $S$ , (2) for every face  $f$  of  $\mathcal{G}$  there exists one site  $s \in S$  such that  $f$  is bounded only by edges which lie on bisectors between  $s$  and other sites of  $S$ , and (3) for every node  $v$  of  $\mathcal{G}$  all edges incident upon  $v$  lie on different weighted bisectors. (Condition (2) is the formal requirement for each face of  $\mathcal{G}$  being defined by exactly one site.) Of course, weighted Voronoi diagrams of  $S$  are examples for bisector graphs of  $S$ .

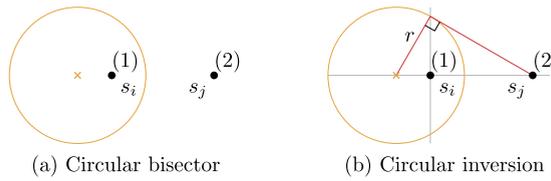
### 1.3. Our contribution

Let  $\mathcal{G}$  be a planar circular-arc graph  $\mathcal{G}$  with  $m$  faces. If the edges of  $\mathcal{G}$  are given by disjoint circles and lines then we can compute all solutions  $(S, \sigma)$  in  $\mathcal{O}(m \log m)$  time such that  $\mathcal{VD}_\sigma(S)$  equals  $\mathcal{G}$ . If  $\mathcal{G}$  has nodes and if all nodes of  $\mathcal{G}$  are of degree three then we can identify in  $\mathcal{O}(m)$  time up to two candidate solutions  $(S, \sigma)$  such that  $\mathcal{G}$  is a weighted bisector graph of the points of  $S$  with weight function  $\sigma$ . We also show that two different inputs  $\mathcal{G}$  may yield the same solution set  $(S, \sigma)$ . Hence, whether or not  $\mathcal{G}$  is an actual Voronoi diagram rather than only a bisector graph for  $(S, \sigma)$  seems difficult to decide without explicitly computing  $\mathcal{VD}_\sigma(S)$ .

## 2. Weighted bisector

For every site  $s_i$  of  $S$  we consider a family of circles  $c_i(t)$  centered at  $s_i$  with radius  $t \cdot \sigma(s_i)$ . Then the (*weighted*) *bisector*  $b(i, j)$  between two distinct sites  $s_i$  and  $s_j$  is given by the trace of the intersection points  $c_i(t) \cap c_j(t)$  for  $r \in \mathbb{R}^+$ :

$$b(i, j) := \{c_i(t) \cap c_j(t) : t \in \mathbb{R}^+\}.$$



**Fig. 3.** (a) The bisector circle (in orange) between the sites  $s_i$  and  $s_j$ . The weights are stated in brackets. (b) Simple construction of  $s_j$  using only  $s_i$  and the circle, based on circular inversion.

For the sake of descriptonal simplicity, we do not explicitly indicate the dependence of a bisector on  $\sigma$ . And, again, in the sequel we will also drop the term “weighted”. It is well-known that the bisector  $b(i, j)$  between two sites  $s_i$  and  $s_j$  forms a circle. (This is easy to see if we recall that ancient Apollonius of Perga showed that a circle is the set of points of a fixed *ratio* of distances to two foci. The two foci in this case are the two input sites, and their bisector is the Apollonian circle which traces out the ratio of their two weights.) Furthermore,  $s_i$  and  $s_j$  lie on a ray that originates at the center of that circle, with one of them on each side of the circle. Aurenhammer and Edelsbrunner [2] state two equations to describe the center and radius of such a bisector circle.

**Lemma 1.** Consider a circle  $C$  with radius  $r$  centered at  $c$  and two distinct sites  $s_i, s_j$  of  $S$  such that  $s_i, s_j$  lie on a ray that originates at  $c$  and such that  $s_i$  lies inside of  $C$  and  $s_j$  lies outside of  $C$ . Let  $r_i$  and  $r_j$  denote the Euclidean distances from  $c$  to  $s_i$  and  $s_j$ . Then the circle  $C$  equals the bisector circle  $b(i, j)$  relative to weights  $\sigma(s_i)$  and  $\sigma(s_j)$  if and only if

$$r_i \cdot r_j = r^2 \quad \text{and} \quad \frac{r}{r_j} = \frac{\sigma(s_i)}{\sigma(s_j)}. \tag{1}$$

**Proof.** We denote the intersection points of  $C$  with the supporting line of the ray from  $c$  to  $s_i$  and  $s_j$  by  $p$  and  $q$ . Assume that  $C$  equals  $b(i, j)$  for appropriate weights  $\sigma(s_i)$  and  $\sigma(s_j)$ . In particular, the points  $p$  and  $q$  are known to lie on  $b(i, j)$ . This implies

$$\frac{r - r_i}{r_j - r} = \frac{\sigma(s_i)}{\sigma(s_j)} = \frac{r + r_i}{r + r_j}.$$

A simple algebraic manipulation yields  $r_i \cdot r_j = r^2$ .

Now assume that  $r_i \cdot r_j = r^2$ . Then

$$\frac{r - r_i}{r_j - r} = \frac{r - r^2/r_j}{r_j - r} = \frac{r}{r_j} \cdot \frac{r - r_j}{r - r_j} = \frac{r}{r_j} = \frac{r}{r_j} \cdot \frac{r_j + r}{r_j + r} = \frac{r + r^2/r_j}{r_j + r} = \frac{r + r_i}{r + r_j}.$$

Therefore, the points  $p, q$  are guaranteed to lie on  $b(i, j)$ . Hence, the line segment  $\overline{pq}$  forms the diameter of  $b(i, j)$  and, thus,  $b(i, j)$  equals  $C$ . Furthermore, appropriate weights fulfill the relation  $r/r_j = \sigma(s_i)/\sigma(s_j)$ .  $\square$

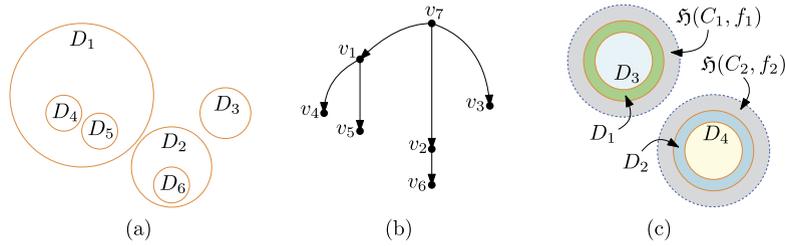
Hence, if  $r_i < r < r_j$  then  $\sigma(s_i) < \sigma(s_j)$ . Note that Equation (1) matches the relation that describes a circular inversion [17]: The inverse  $\mathcal{H}(C, p)$  of a point  $p$  about a circle  $C$  with center  $c$  and radius  $r$  is a point  $p'$  which lies on the ray from  $c$  through  $p$  such that  $d(c, p) \cdot d(c, p') = r^2$ . Let  $\mathcal{H}(C, A)$  denote the circular inversion of the area  $A \subseteq \mathbb{R}^2$  about the circle  $C$ . Then Lemma 1 can be re-phrased as follows: Two sites  $s_i, s_j \in S$  have a circle  $C$  as their bisector circle (for appropriate weights) if and only if  $s_i = \mathcal{H}(C, s_j)$ , and vice versa. This insight establishes the following lemma.

**Lemma 2.** Let  $\mathcal{G}$  be a bisector graph of a set  $S$  of weighted point sites. If  $\mathcal{G}$  contains a full circle  $C$  as an edge then at least one site of  $S$  lies within the circular disk bounded by  $C$  such that  $C$  is its bisector circle with some other site of  $S$  outside of  $C$ .

The theory of circular inversion tells us that the inversion of a circle that is inside of  $C$  and passes through its center  $c$  is a line, while all other circles inside of  $C$  invert to circles outside of  $C$ . The interior of a circular disk  $D$  bounded by a circle  $C'$  maps to the interior of the inversion of  $C'$  if  $D$  does not contain  $c$ , and to its exterior otherwise. The center  $c$  of  $C$  is mapped to a point at infinity and vice versa. Furthermore, as sketched in Fig. 3, there is a simple way to construct  $s_j$  if  $C$  and  $s_i$  are known. Hence, given a circle with radius  $r$  and center  $c$ , we may choose any point  $p$  inside or outside of the circle and obtain the inverse point  $p'$  using the equation  $\overrightarrow{cp} \cdot \overrightarrow{cp'} = r^2$ , where  $\overrightarrow{uv}$  denotes the vector from  $u$  to  $v$  and the dot stands for the dot product of two vectors.

Now interpret  $\mathbb{R}^2$  as the complex plane  $\mathbb{C}$ . A Möbius transformation of  $\mathbb{C}$  is a non-constant rational function that maps  $z \in \mathbb{C}$  to

$$\frac{az + b}{cz + d} \quad \text{for } a, b, c, d \in \mathbb{C} \text{ with } ad - bc \neq 0.$$



**Fig. 4.** (a) An example illustration of  $\mathcal{G}$  containing nested non-intersecting circles; (b) The dual graph  $\mathcal{D}$  of  $\mathcal{G}$  in (a); (c) An example where there is no solution as  $\mathfrak{S}(C_1, f_1) \cap \mathfrak{S}(C_2, f_2) = \emptyset$ .

By setting  $a = d := 0$  and  $b = c := 1$  we obtain an inversion (and reflection about the real axis) of a point  $z \in \mathbb{C}$  about the unit circle centered at the origin. More generally, the theory of Möbius transformations tells us that every circular inversion in  $\mathbb{C}$  can be modeled as a (potentially conjugated) Möbius transformation. Every such Möbius transformation maps circles and lines to circles and lines. Furthermore, the composition of two Möbius transformations yields yet another Möbius transformation, which can be computed by multiplying two matrices of  $GL_2(\mathbb{C})$ , the so-called general linear group of invertible  $2 \times 2$  matrices over  $\mathbb{C}$ . Reference is given to Schwerdtfeger [18] for a standard introduction to Möbius transformations.

We conclude this section with a definition used in the sequel: We say that a (non-empty) set  $X \subset \mathbb{R}^2$  is *nested* inside a set  $Y \subset \mathbb{R}^2$  if  $\mathbb{R}^2 \setminus Y$  has a bounded connected component  $Z$  such that  $X \subseteq Z$ .

### 3. Non-intersecting circles and lines

#### 3.1. No nested circles in $\mathcal{G}$

In this section we employ Lemma 1 to recognize specific types of input graphs  $\mathcal{G}$  and to reconstruct suitable weighted point sets  $(S, \sigma)$  such that  $\mathcal{G}$  equals  $\mathcal{VD}_\sigma(S)$ . Let  $\mathcal{G}$  be a collection of  $m - 1$  circles  $C_1, C_2, \dots, C_{m-1}$  which do not intersect pairwise and which are not nested. Then  $\mathcal{G}$  partitions the plane into  $m - 1$  circular disks  $D_1, D_2, \dots, D_{m-1}$  and one unbounded region. Lemmas 1 to 2 imply that the highest-weighted site  $s_m \in S$  has to lie in the unbounded region and that each disk  $D_i$  has to contain a corresponding site  $s_i$ . Hence, we get  $|S| = m$ .

We choose an arbitrary point in the unbounded region as site  $s_m$  of  $S$ . We may also choose its weight arbitrarily. Based on Lemma 1 we obtain point sites  $s_1, \dots, s_{m-1}$  of  $S$ , with  $s_i$  inside of  $C_i$ . The weights of the remaining sites are thus fixed since the inversion property needs to hold; cf. Lemma 1. In other words, the site  $s_m$  is chosen within  $\bigcap_{1 \leq i \leq m-1} \mathfrak{S}(C_i, D_i)$ . By construction,  $s_m$  is the highest-weighted site and its Voronoi region is the unbounded face. Furthermore, due to Lemma 1, we know that  $C_i$  forms the bisector circle  $b(i, m)$  for  $i \in \{1, 2, \dots, m - 1\}$ .

Suppose that  $\mathcal{VD}_\sigma(S)$  differs from  $\mathcal{G}$ . Since every site of  $S$  needs to have its own Voronoi region,  $\mathcal{VD}_\sigma(S)$  cannot be a genuine subset of  $\mathcal{G}$ . Hence there exist  $1 \leq i < j < m$  such that  $\mathcal{VD}_\sigma(S)$  contains a point  $p$  on the bisector  $b(i, j)$  which does not lie on a circle of  $\mathcal{G}$ . Such a point  $p$  cannot lie within both  $D_i$  and  $D_j$ . W.l.o.g.,  $p$  does not lie within  $D_i$ . Then  $d_\sigma(p, s_m) < d_\sigma(p, s_i)$  and, thus,  $p \notin \mathcal{VD}_\sigma(S)$ . We conclude that  $\mathcal{VD}_\sigma(S)$  equals  $\mathcal{G}$ . We summarize this result in the following theorem.

**Theorem 1.** *If  $\mathcal{G}$  is given by a collection of  $m - 1$  circles that are not nested and do not intersect pairwise then  $\mathcal{G}$  always admits (a family of) solutions  $(S, \sigma)$  such that  $\mathcal{G}$  equals  $\mathcal{VD}_\sigma(S)$ . A sample solution  $(S, \sigma)$  can be determined in  $\mathcal{O}(m)$  time.*

#### 3.2. Nested circles in $\mathcal{G}$

Let  $\mathcal{G}$  be a collection of  $m - 1$  circles  $C_1, C_2, \dots, C_{m-1}$  which do not intersect pairwise. They are allowed to be nested, though. Again we denote the disks defined by these circles by  $D_1, D_2, \dots, D_{m-1}$ . For  $1 \leq i \leq m - 1$ , we denote by  $f_i$  the face of  $\mathcal{G}$  inside of  $D_i$  that is bounded by  $C_i$  and, possibly, some other circles nested inside of  $C_i$ . The unbounded face is given by  $f_m$  and its “disk” is denoted by  $D_m$ . Lemma 2 again implies that every disk has to contain its defining site and we get  $|S| = m$ . However, since the circles may be nested, it is no longer good enough to choose  $s_m$  within  $\bigcap_{1 \leq i \leq m-1} \mathfrak{S}(C_i, D_i)$ .

We construct a dual graph  $\mathcal{D}$  of  $\mathcal{G}$  in the following way: For every face  $f_i$  we create a node  $v_i$  in  $\mathcal{D}$ . Two nodes in  $\mathcal{D}$  are connected by an edge if their faces in  $\mathcal{G}$  are adjacent. Since the circles in  $\mathcal{G}$  are non-intersecting,  $\mathcal{D}$  can not contain a cycle but forms a tree. We turn  $\mathcal{D}$  into a rooted tree by rooting it at the node that corresponds to the unbounded face. In Figs. 4a and 4b we illustrate such a setup where  $v_7$  is the root node that corresponds to the unbounded face.

Next we define a *solution set*,  $ss(f)$ , for each face  $f$  of  $\mathcal{G}$  recursively as the loci of points that are feasible for a point site inside of  $f$ .

$$ss(f_i) := \begin{cases} D_i & \text{if } v_i \text{ is a leaf of } \mathcal{D}, \\ D_i \cap \left\{ \bigcap_{v_j \text{ is a child of } v_i} \mathfrak{S}(C_j, ss(f_j)) \right\} & \text{otherwise.} \end{cases}$$

In particular, the solution set  $ss(f_m)$  for the unbounded face  $f_m$  of  $\mathcal{G}$  is obtained by starting at the root node of  $\mathcal{D}$  and following all branches until the leaves of  $\mathcal{D}$  are reached. Then the respective point sets are mapped back to the unbounded face.

Let  $s_i$  lie within  $f_i$ . Then an inductive proof immediately implies that it is necessary for  $s_i$  to lie within  $ss(f_i)$  for all  $1 \leq i \leq m$ . Hence, if  $ss(f_i)$  is empty for some  $1 \leq i \leq m$  then there exists no solution  $(S, \sigma)$  such that  $\mathcal{VD}_\sigma(S)$  matches  $\mathcal{G}$ ; cf. Fig. 4c. (However, in general  $\mathcal{G}$  could still be a bisector graph for suitable  $(S, \sigma)$ .) Otherwise, we can choose an arbitrary point  $s_m$  within  $ss(f_m)$ . We may also choose its weight arbitrarily. The positions of all other sites  $s_1, \dots, s_{m-1}$  (and appropriate weights) are obtained by recursively computing circular inversions of  $s_m$ , as implied by  $\mathcal{D}$ . It remains to argue that this construction is sufficient to ensure that  $\mathcal{VD}_\sigma(S)$  matches  $\mathcal{G}$ .

**Lemma 3.** *If  $(S, \sigma)$  is obtained by randomly picking a site  $s_m \in ss(f_m)$  and computing all  $s_1, \dots, s_{m-1}$  by circular inversions, as outlined above, then  $\mathcal{VD}_\sigma(S)$  matches  $\mathcal{G}$ .*

**Proof.** Due to Lemma 1, we know that  $C_i$  forms the bisector circle  $b(i, j)$  if  $s_j$  lies outside of  $D_i$  and if  $s_i$  is obtained by a circular inversion of  $s_j$  about  $C_i$ . So suppose that there exist  $1 \leq i < j \leq m$  such that  $\mathcal{VD}_\sigma(S)$  contains a point on the bisector  $b(i, j)$  which does not lie on a circle of  $\mathcal{G}$ . The arguments used for non-nested circles imply that this could only happen if the node  $v_j$  is an ancestor of the node  $v_i$  (or vice versa). Since, by construction,  $\mathcal{G}$  contains the bisector of  $s_i$  and  $s_j$  if  $v_i$  is a child of  $v_j$ , we know that there is at least one node  $v_k$  that is a child of  $v_j$  and an ancestor of  $v_i$ . Hence,  $\mathcal{R}_\sigma(s_i, S) \subseteq D_i \subset D_k \subset D_j$  and  $D_k$  equals  $b(k, j)$ . However, then every point of  $\mathcal{R}_\sigma(s_i, S)$  is closer to  $s_k$  than to  $s_j$ , making it impossible for  $s_i$  and  $s_j$  to share a point that belongs to  $\mathcal{VD}_\sigma(S)$ .  $\square$

The solution set  $ss(f_m)$  is described by  $m - 1$  circles (or straight lines) together with sidedness information that tells us on which side of a circle (or line) the feasible points lie. If  $ss(f_m)$  is not empty then it forms a face in the arrangement of the  $m - 1$  circles (and lines). Note that this approach works also if the circles of  $\mathcal{D}$  are allowed to touch each other. We now focus on the actual computation of  $ss(f_m)$ .

**Lemma 4.** *In  $\mathcal{O}(m)$  time we can obtain appropriate transformations of the  $m - 1$  disks associated with all nodes of  $\mathcal{D}$  except for its root node.*

**Proof.** As outlined in Section 2, we interpret  $\mathbb{R}^2$  as the complex plane  $\mathbb{C}$  and proceed as follows: For every non-leaf node of  $\mathcal{D}$  we set up the appropriate Möbius transformation. (The identity transformation is used for the root of  $\mathcal{D}$ .) Then we apply an in-order traversal to  $\mathcal{D}$  and, for each non-leaf node  $v$  other than the root of  $\mathcal{D}$ , we compute the composition of the Möbius transformation stored at  $v$  with the Möbius transformation stored at the parent of  $v$ . This composed Möbius transformation replaces the old transformation stored at  $v$ . Hence, in  $\mathcal{O}(m)$  time we can obtain appropriate transformations of the  $m - 1$  disks associated with all nodes of  $\mathcal{D}$  except for its root node.  $\square$

In order to obtain  $ss(f_m)$  it remains to compute the intersection of these disks. The intersection of  $m - 1$  disks or the disks' complements can be constructed in  $\mathcal{O}(m \log m)$  time. Brown describes this representation in detail in his thesis [19]. Aurenhammer and Edelsbrunner [2] use an extended version to construct the weighted Voronoi diagram. We follow their description and describe in the following how we apply it to our setting.

**Lemma 5.** *The intersection of the  $m - 1$  disks or the disks' complements at the root node of  $\mathcal{D}$  can be computed in  $\mathcal{O}(m \log m)$  time.*

**Proof.** For every disk we embed its defining circle in the  $xy$ -plane in  $\mathbb{R}^3$ . We choose an arbitrary point of inversion  $p_i$  in  $\mathbb{R}^3$  that does not lie on the  $xy$ -plane, e.g.,  $(0, 0, 1)$  the point above the origin at  $z = 1$ . Note that by using a single circle and a point not on the circle we can uniquely define a sphere such that both circle and point lie on the sphere's boundary. Hence, for each circle we create a unique sphere in combination with  $p_i$ . Then,  $p_i$  lies on all  $m - 1$  spheres. Using  $p_i$  as point of inversion we apply a spherical inversion that creates a half-space from every sphere. For each disk computed for  $ss(f_m)$  we know whether the disk or its complement is to be considered. If the disk's complement is required then we form the complement of the respective half-space. We can form the intersection of these  $m - 1$  half-spaces in  $\mathcal{O}(m \log m)$  time. The result is a convex polyhedron  $\mathfrak{P}$  in  $\mathbb{R}^3$ . To obtain a representation of  $\mathbb{R}^2$  we invert the  $xy$ -plane using  $p_i$  as well. The result is a sphere  $S_{xy}$  that contains  $p_i$ . Then we intersect  $\mathfrak{P}$  with  $S_{xy}$ . Since  $\mathfrak{P}$  is formed from  $m - 1$  half-spaces it has a combinatorial complexity of  $\mathcal{O}(m)$ . Hence, we can traverse each facet  $f$  of  $\mathfrak{P}$  and intersect it with  $S_{xy}$ . As each facet is convex as well we find the intersection for every  $f$  in  $\mathcal{O}(|f|)$  time. We keep the portion of each facet that lies outside of  $S_{xy}$ . Hence,  $ss(f_m)' := \mathfrak{P} \cap \overline{S_{xy}}$  is constructed in  $\mathcal{O}(m)$  time, where  $\overline{S_{xy}}$  denotes the complement of  $S_{xy}$ . Let  $e'$  denote an edge of  $\mathfrak{P}$  that is shortened by the intersection process. Let  $e$  denote  $e'$  transformed back to the  $xy$ -plane. The two planes that are locally incident at  $e'$  imply two specific disks with respect to  $e$  in the  $xy$ -plane. The shortened endpoint of  $e'$ , and  $e$  respectively, is the point where the boundaries of the two disks meet.

Transforming  $ss(f_m)'$  back into the  $xy$ -plane yields  $ss(f_m)$  and is accomplished in linear time in the size of the intersection. Therefore we obtain  $ss(f_m)$  in overall  $\mathcal{O}(m \log m)$  time.  $\square$

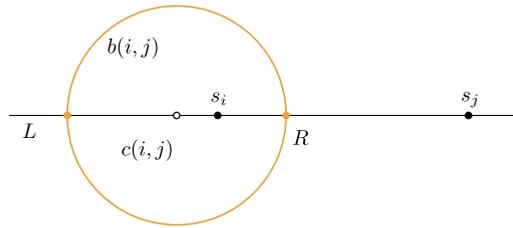


Fig. 5. Two weighted sites and their bisector.

Recall that our circular inversions may create half-planes as well: In case a transformed circle intersects the center of an inversion circle it is transformed into a line, i.e., half-plane. Let  $\ell$  denote such a line in  $\mathbb{R}^2$ . Instead of a sphere we form a plane in  $\mathbb{R}^3$  that intersects  $\ell$  and the inversion point  $p_i$ . We take advantage of the inversion property of the plane that intersects the inversion point, that is, the plane inverts into the same plane. The half-plane defining  $\ell$  defines the half-space for the plane and we can apply our half-space intersection as described above. Theorem 2 summarizes the main result of this section.

**Theorem 2.** Let  $\mathcal{G}$  be a collection of  $m - 1$  circles  $C_1, C_2, \dots, C_{m-1}$  that do not intersect pairwise. Then we can compute all solutions  $(S, \sigma)$  such that  $\mathcal{G}$  matches  $\mathcal{VD}_\sigma(S)$  in  $\mathcal{O}(m \log m)$  time.

### 3.3. Nested circles and lines in $\mathcal{G}$

Let  $\mathcal{G}$  be a collection of  $m - 1$  circles and lines which do not intersect pairwise. The circles are allowed to be nested, though. We follow the notation of the previous section. Let  $k$  denote the number of lines  $\ell_1, \dots, \ell_k$  in  $\mathcal{G}$ . Clearly for  $k > 1$  the lines have to be parallel to be non-intersecting. Since a line is not finite it can only partition the unbounded face. Hence, the lines partition the unbounded face into  $k + 1$  unbounded regions. For two sites  $s_i, s_j$  to form a line  $\ell \in \mathcal{G}$  as their bisector the sites must have equal weight and therefore equal distance to  $\ell$ , cf. Section 2. We modify our inversion function  $\mathfrak{H}(\cdot, \cdot)$  such that the inverse of a point about a line is simply its mirrored image. To obtain a solution we construct the dual graph  $\mathcal{D}$  for each unbounded face separately. Then, we compute the solution set for each tree root  $v^1, \dots, v^{k+1}$ . Finally, starting at an unbounded face that is incident to at most one line  $\ell$ , we map its solution arrangement via  $\ell$  to the next consecutive face and form the intersection with its solution space. We repeat this process until we reach the last unbounded face and thereby obtain a full characterization of the solution.

## 4. Recognizing $\mathcal{G}$ as bisector graph

Given a planar circular-arc graph  $\mathcal{G}$ , we ask whether  $\mathcal{G}$  is a weighted bisector graph. If it is a bisector graph, then we seek a set  $S$  of suitable sites and a corresponding weight function  $\sigma$ . That is, we want to find a solution  $(S, \sigma)$  such that every edge of  $\mathcal{G}$  lies on a bisector defined by two sites of  $S$ . Contrary to Section 3 we now assume that  $\mathcal{G}$  contains at least one node. For a start, we also assume that all nodes of  $\mathcal{G}$  are of degree exactly three.

We begin with studying the structure of weighted bisector graphs. As a start we recall Condition (3) in the definition of a bisector graph (at the end of Section 1.2): The three edges of  $\mathcal{G}$  that are incident at a degree-three node  $v$  of  $\mathcal{G}$  have to lie on three distinct bisector circles. Hence, as a first check, we scan all nodes of  $\mathcal{G}$  to verify that this prerequisite is met. If it is not met then we do already know that  $\mathcal{G}$  is no bisector graph. Of course, this scan can be carried out in time linear in the number of nodes and edges of  $\mathcal{G}$ .

**Lemma 6.** Let  $s_i, s_j, s_k$  denote three sites, with  $\sigma(s_i) < \sigma(s_j) < \sigma(s_k)$ . Then there exists a line  $\ell$  which contains the centers of all three bisectors  $b(i, j)$ ,  $b(j, k)$ , and  $b(i, k)$ .

**Proof.** Consider two sites  $s_i$  and  $s_j$  with  $\sigma(s_i) < \sigma(s_j)$ . The center of their common bisector circle  $b(i, j)$  is denoted by  $c(i, j)$ ; cf. Fig. 5.

The points  $L$  and  $R$  are two points on the bisector, equidistant (in weighted terms) to both  $s_i$  and  $s_j$ . Point  $L$  is of maximal distance and point  $R$  of minimal distance. The distance relation  $(R - s_i) / \sigma(s_i) = (s_j - R) / \sigma(s_j)$  yields

$$R = \frac{s_i \sigma(s_j) + s_j \sigma(s_i)}{\sigma(s_i) + \sigma(s_j)}.$$

Similarly, the relation  $(s_i - L) / \sigma(s_i) = (s_j - L) / \sigma(s_j)$  yields

$$L = \frac{s_j \sigma(s_i) - s_i \sigma(s_j)}{\sigma(s_i) - \sigma(s_j)}.$$

Since  $c(i, j) = 1/2(L + R)$ , we immediately see that

$$c(i, j) = \frac{s_j\sigma(s_i)^2 - s_i\sigma(s_j)^2}{\sigma(s_i)^2 - \sigma(s_j)^2}.$$

Now consider three sites,  $s_i, s_j, s_k$  with  $\sigma(s_i) < \sigma(s_j) < \sigma(s_k)$ . We want to show that the centers of their bisector circles, i.e.,  $c(i, j)$ ,  $c(i, k)$ , and  $c(j, k)$  are collinear. Three points are collinear if and only if the area of the triangle defined by them is zero. Thus, the three points in question are on the same supporting line if the following determinant vanishes:

$$D := \begin{vmatrix} x(c(i, j)) & y(c(i, j)) & 1 \\ x(c(i, k)) & y(c(i, k)) & 1 \\ x(c(j, k)) & y(c(j, k)) & 1 \end{vmatrix}$$

where  $x(p)$  and  $y(p)$  denote the  $x$ - and  $y$ -coordinates of a point  $p$ .

After expanding the determinant we get

$$\begin{aligned} D &= x(c(i, j)) \cdot (y(c(i, k)) - y(c(j, k))) \\ &\quad + x(c(i, k)) \cdot (y(c(j, k)) - y(c(i, j))) \\ &\quad + x(c(j, k)) \cdot (y(c(i, j)) - y(c(i, k))) \\ &= \frac{(x(s_j)\sigma(s_i)^2 - x(s_i)\sigma(s_j)^2) \left( \frac{y(s_k)\sigma(s_i)^2 - y(s_i)\sigma(s_k)^2}{\sigma(s_i)^2 - \sigma(s_k)^2} - \frac{y(s_k)\sigma(s_j)^2 - y(s_j)\sigma(s_k)^2}{\sigma(s_j)^2 - \sigma(s_k)^2} \right)}{\sigma(s_i)^2 - \sigma(s_j)^2} \\ &\quad + \frac{(x(s_k)\sigma(s_i)^2 - x(s_i)\sigma(s_k)^2) \left( \frac{y(s_k)\sigma(s_j)^2 - y(s_j)\sigma(s_k)^2}{\sigma(s_j)^2 - \sigma(s_k)^2} - \frac{y(s_j)\sigma(s_i)^2 - y(s_i)\sigma(s_j)^2}{\sigma(s_i)^2 - \sigma(s_j)^2} \right)}{\sigma(s_i)^2 - \sigma(s_k)^2} \\ &\quad + \frac{(x(s_k)\sigma(s_j)^2 - x(s_j)\sigma(s_k)^2) \left( \frac{y(s_j)\sigma(s_i)^2 - y(s_i)\sigma(s_j)^2}{\sigma(s_i)^2 - \sigma(s_j)^2} - \frac{y(s_k)\sigma(s_i)^2 - y(s_i)\sigma(s_k)^2}{\sigma(s_i)^2 - \sigma(s_k)^2} \right)}{\sigma(s_j)^2 - \sigma(s_k)^2} \\ &= 0, \end{aligned}$$

thus having proved the claim.  $\square$

**Lemma 7.** Let  $s_i, s_j, s_k$  denote three distinct sites. If two of the three bisectors  $b(i, j)$ ,  $b(j, k)$ , and  $b(i, k)$  intersect at a point  $p$ , then all three bisectors intersect in  $p$ .

**Proof.** Assume that  $b(i, j)$  and  $b(j, k)$  intersect in the point  $p$ . Since  $b(i, j)$  is the set of points of equal (weighted) distance to  $s_i$  and  $s_j$ , and  $b(j, k)$  is the set of points of equal distance to  $s_j$  and  $s_k$ , it follows that  $p$  has equal distance to  $s_i, s_j$ , and  $s_k$ . Thus,  $p$  also lies on  $b(i, k)$ .  $\square$

**Corollary 1.** Let  $s_i, s_j, s_k$  denote three distinct sites. Then  $b(i, j)$ ,  $b(j, k)$  and  $b(i, k)$  intersect pairwise either in exactly two points, exactly one point, or not at all.

#### 4.1. Finding sites from bisectors

Let  $v$  denote a node of degree three of the graph  $\mathcal{G}$ . This node is the intersection of three circular arcs, i.e., of the bisector circles of three sites. We seek the locations of these sites given the bisector arcs. Let  $i, j, k$  be the indices of these three sites, and denote the centers of their bisector circles by  $c(i, j)$ ,  $c(j, k)$ ,  $c(i, k)$ ; see Fig. 6.

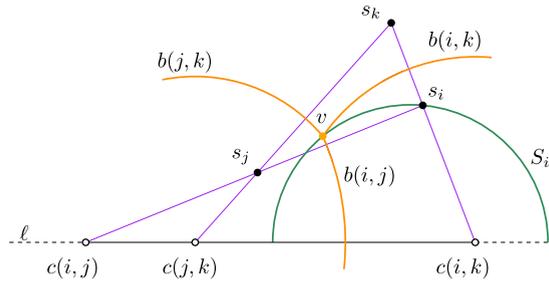
Lemma 6 tells us that the centers  $c(i, j)$ ,  $c(j, k)$ ,  $c(i, k)$  lie on a line  $\ell$ . Thus, we can, w.l.o.g., assume the centers to lie on the  $x$ -axis and  $v$  to be at coordinates  $(0, 1)$ . (The general case is reduced to this setting by rotation, scaling, and translation.)

Next, we set up the conjugated Möbius transforms  $\mathfrak{H}_{i,j}$ ,  $\mathfrak{H}_{j,k}$ , and  $\mathfrak{H}_{i,k}$ , across bisectors  $b(i, j)$ ,  $b(j, k)$ , and  $b(i, k)$ , respectively. Furthermore, we concatenate all three transforms into a single transform  $\mathfrak{H} := \mathfrak{H}_{i,j} \circ \mathfrak{H}_{j,k} \circ \mathfrak{H}_{i,k}$ .

Let the  $x$ -coordinates of  $c(i, j)$ ,  $c(i, k)$ , and  $c(j, k)$  be given by  $x_1, x_2$ , and  $x_3$ . Then, the Möbius transform  $\mathfrak{H}$  is given by the matrix product of three individual transforms. Simple math yields

$$\mathfrak{H} = \begin{pmatrix} (x_1 - x_2 + x_3 + x_1x_2x_3) & (1 + x_1x_2 - x_1x_3 + x_2x_3) \\ (1 + x_1x_2 - x_1x_3 + x_2x_3) & (-x_1 + x_2 - x_3 - x_1x_2x_3) \end{pmatrix}.$$

Now it must hold that  $S_i = \mathfrak{H}_{i,j}(S_j)$  and  $S_j = \mathfrak{H}_{j,k}(S_k)$  and  $S_k = \mathfrak{H}_{i,k}(S_i)$ , where  $S_i$  is a solution set for site  $s_i$ , and  $S_j$  and  $S_k$  are defined likewise. In particular,  $S_i = \mathfrak{H}_{i,j}(\mathfrak{H}_{j,k}(\mathfrak{H}_{i,k}(S_i)))$ , or equivalently,  $S_i = \mathfrak{H}(S_i)$ .



**Fig. 6.** Reconstructing the three sites that traced out the orange bisector arcs incident at node  $v$ . A site  $s_i$  necessarily lies on an arbitrary location on the green solution circle  $S_i$ . Sites  $s_j$  and  $s_k$  then follow by inversions of  $s_i$  across the corresponding bisector.

Solving this equation for  $S_i =: (x, y)$  yields

$$y^2 + x^2 - \frac{2(x_1 - x_2 + x_3 + x_1x_2x_3)}{1 + x_1x_2 - x_1x_3 + x_2x_3}x - 1 = 0,$$

which is a circle with center at coordinates

$$c(S_i) = \left( \frac{x_1 - x_2 + x_3 + x_1x_2x_3}{1 + x_1x_2 - x_1x_3 + x_2x_3}, 0 \right)$$

and which goes through the intersection point of all bisectors at  $(0, 1)$ . Likewise, the solution set for  $S_j$  and  $S_k$  are circles, namely the corresponding inverses or Möbius transforms of  $S_i$ .

Therefore, given three intersecting bisectors, we can find sites  $s_i, s_j, s_k$  that generate these bisectors. In general the sites are not uniquely defined, and instead each site lies on a “solution circle”. This establishes Lemma 8.

**Lemma 8.** *Let  $v$  denote a node of degree three of  $\mathcal{G}$ . Let  $b(i, j), b(i, k),$  and  $b(j, k)$  define the three arcs that meet at  $v$ . Then the sites  $s_i, s_j,$  and  $s_k$  lie on solution circles  $S_i, S_j,$  and  $S_k$ , which can be constructed. Picking a specific locus for  $s_i$  on  $S_i$  automatically fixes the other sites, and vice versa.*

#### 4.2. Bisector graph recognition

In the following, we describe how to obtain a solution  $(S, \sigma)$  such that every edge of  $\mathcal{G}$  lies on one bisector defined by  $(S, \sigma)$ , if such a solution exists, and thereby detects whether  $\mathcal{G}$  is a bisector graph. Recall that every node of  $\mathcal{G}$  has degree three. (In Section 4.4 we waive this restriction and discuss the general case.)

Let  $f_1, \dots, f_m$  denote the faces of  $\mathcal{G}$ , where  $m \geq n$  for  $n := |S|$ . Let  $f_i$  denote a face of  $\mathcal{G}$  with a maximal number of boundary nodes. The nodes on the boundary of  $f_i$  are given by  $v_1, \dots, v_k$ . We have  $k > 0$  because we had assumed to have at least one node in  $\mathcal{G}$ . As each node is given by the intersection of distinct bisectors, we necessarily have  $k > 1$ . In case  $k = 2$ , we apply Lemma 8 to the two degree-three nodes of  $f_i$ , thus obtaining a family of solutions for  $f_i$ . These local solutions are combined by means of the process outlined in Section 3.2.

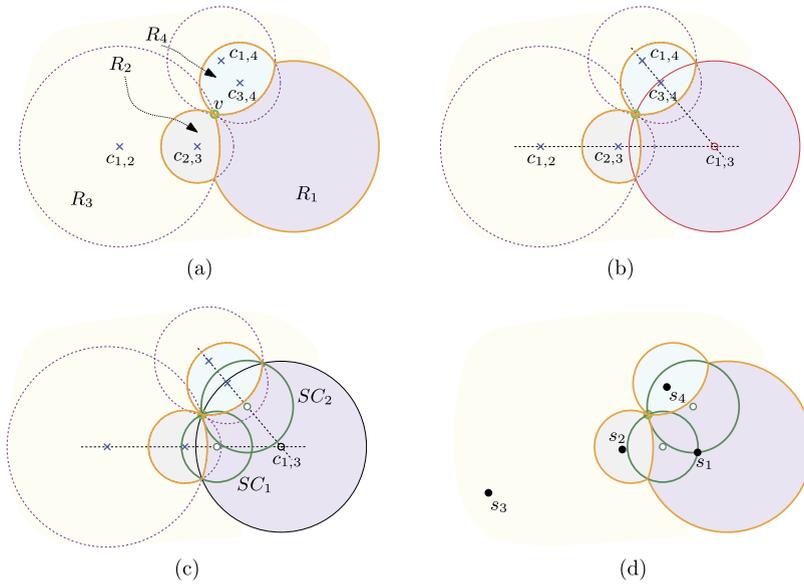
Otherwise, the face  $f_i$  has  $k \geq 3$  nodes. Note that it is still possible for each bounded face to have at most two nodes. However, in that case two or more nodes will be incident to the unbounded face. Applying Lemma 8 to one node of  $f_i$  will yield a family of solutions for the site  $s_i$  for  $f_i$ . However, we can even restrict this solution to a constant number of points as follows. We traverse the boundary of  $f_i$ , node by node, and apply Lemma 8. Thereby we produce  $k$  solution circles for  $s_i$ , denoted by  $S_i^1, \dots, S_i^k$ . Necessarily,  $s_i$  is in the intersection of these solution circles:  $s_i = \bigcap_{j=1}^k S_i^j$ .

Let  $S_i^1$  and  $S_i^2$  denote two circles such that  $S_i^1 \neq S_i^2$ . Note that a solution circle  $S_i^j$  computed for  $v_j$  intersects  $v_j$ . Thus, for every  $S_i^j$  we can find a different node (when  $k \geq 3$ ) on  $f_i$  which is not on  $S_i^j$  and so its solution circle will be different. Therefore, two such circles exist. Hence, the intersection  $s_i = \bigcap_{j=1}^k S_i^j$  contains at most two points. If  $s_i = \emptyset$  then  $\mathcal{G}$  is not a bisector graph. Otherwise, these points are loci for the site, thus establishing Lemma 9.

**Lemma 9.** *For a face  $f$  of  $\mathcal{G}$  with at least three boundary nodes the set of solutions for the site of  $f$  consists of at most two points.*

Every other solution circle intersects either both solution points or reduces the solution to a single point  $p$ . We now assume that the solution consists of a single point  $p$ . (Otherwise we apply the following process to both points.)

We choose  $p$  as locus for our site  $s_i$  with arbitrary weight. We use the identified site  $s_i$  and apply inversions  $\mathfrak{S}_i(\cdot)$  via every edge defining  $f_i$  to obtain all neighboring sites. Then we repeat the process for all neighboring sites and their neighbors in turn, traversing the entire graph  $\mathcal{G}$ , breadth first. Thereby we obtain the solution  $(S, \sigma)$  after  $\mathcal{O}(m)$  inversions. In every face processed we verify that the solution circles defined by the boundary nodes contain the alleged site. If a solution circle does not contain the alleged site then  $\mathcal{G}$  does not constitute a bisector graph for this starting point  $p$ .



**Fig. 7.** In a bisector graph with nodes of higher degree, we can partition the incident arcs, construct individual solution sets, and intersect those. (a) Graph  $\mathcal{G}$  with centers of the bisector arcs incident at  $v$ . For instance,  $c_{1,2}$  is the center of the bisector between sites  $s_1$  and  $s_2$ , which had traced out regions  $R_1$  and  $R_2$ . (b) Even if we do not have the bisector given, we can construct the bisector (red) between  $s_1$  and  $s_3$ . (c) Now we have two systems of three bisectors each: The  $s_1, s_2, s_3$ -system, and the  $s_1, s_3, s_4$ - system. For each of them, we can construct solution circles for  $s_1$  (green):  $SC_1$  and  $SC_2$ . (d) The intersection of these solution circles yields a position for  $s_1$ . The other sites are obtained by standard spherical inversions across the appropriate bisector arcs.

4.3. Complexity

If  $k = 2$  then we can find all local solutions in  $\mathcal{O}(1)$  time. We note that we are able to handle nested circles in  $\mathcal{O}(m)$  time whenever  $\mathcal{G}$  includes degree-three nodes (as assumed) because applying the conjugated Möbius transform on  $\mathcal{G}$  already yields all feasible sites for the subgraphs of  $\mathcal{G}$  that contain nodes. Thus, it is not necessary to compute the entire solution set for the nested circles in  $\mathcal{G}$ .

If  $k \geq 3$  then we find a face  $f_i$  with  $k$  nodes in  $\mathcal{O}(m)$  time. We find the solution circles in  $\mathcal{O}(1)$  time per node and intersect them in  $\mathcal{O}(k)$  time. Using the resulting intersection point  $p$ , we apply the breadth-first traversal, which takes again  $\mathcal{O}(m)$  time, as each face  $f$  is processed in time linear in the combinatorial size of the face. Therefore we obtain Theorem 3.

**Theorem 3.** Given a planar circular-arc graph  $\mathcal{G}$  with  $m$  faces, in time  $\mathcal{O}(m)$  we can detect whether  $\mathcal{G}$  constitutes a bisector graph and, if yes, determine the (at most two) suitable solutions  $(S, \sigma)$ .

Note that  $S$  may contain more than one site on the same locus. This is admissible for a bisector graph but not for a Voronoi diagram.

4.4. Handling nodes of higher degree

At the start of this section we had assumed that all nodes of  $\mathcal{G}$  are of degree three. This restriction can be waived while still maintaining Theorem 3. Consider a node of  $\mathcal{G}$  with degree larger than three and its incident bisector circular arcs. We can distinguish two fundamentally distinct cases: a) The centers of all the circles are collinear, or b) not all of them are collinear.

If all centers lie on the same supporting line then we can construct a Möbius transform  $\mathcal{H}$  that represents the inversions across all the incident bisector arcs in the same way as described in the proof of Lemma 8. Once we have that, we can once more solve  $s_i = \mathcal{H}(s_i)$  and thus obtain all valid locations for a site. Note that, for instance with degree-four nodes and the bisector circles in specific configurations, it may be that the entire plane is a valid solution. Other degree-four configurations may yield only the trivial solution of the intersection point of all bisectors and thus will never appear in bisector graphs.

If not all centers lie on the same supporting line then we can reduce the problem to subproblems of smaller size. For instance, if we have a degree-four node, we consider appropriate pairs of centers of the bisector arcs. For each pair, we construct its supporting line, and its intersection is the center for another bisector, one that did not have arcs represented in  $\mathcal{G}$ . Thus, for each supporting line we now have three bisector arcs, and the procedure from Lemma 8 yields a solution circle for a site. By intersecting the solution circles for the different supporting lines we obtain the location of a site. Fig. 7 demonstrates this procedure.

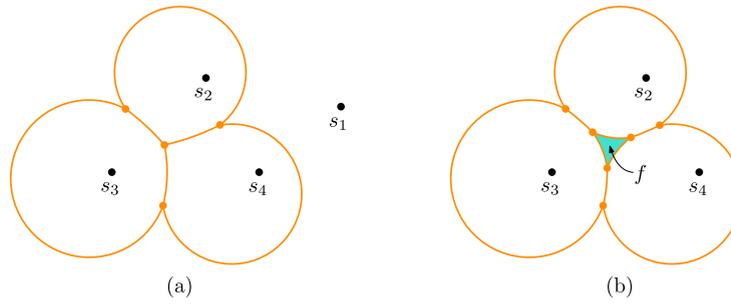


Fig. 8. Given sites  $S := \{s_1, \dots, s_4\}$  and an appropriate weight function  $\sigma$ , then (a) shows a bisector graph of  $(S, \sigma)$  and (b) shows  $\mathcal{VD}_\sigma(S)$ .

#### 4.5. Recognizing $\mathcal{G}$ as Voronoi diagram

Consider a planar circular-arc graph  $\mathcal{G}$ . Does there exist a solution set  $(S, \sigma)$  such that  $\mathcal{VD}_\sigma(S)$  equals  $\mathcal{G}$ ? Since every Voronoi diagram is a bisector graph, we start by applying the bisector-graph detection presented in the previous section.

If  $\mathcal{G}$  is not recognized as a bisector graph, then it is not a Voronoi diagram. If  $\mathcal{G}$  does not contain nodes then  $\mathcal{G}$  is guaranteed to match  $\mathcal{VD}_\sigma(S)$  for all solutions  $(S, \sigma)$  returned by our reconstruction algorithm. In the general case, where  $\mathcal{G}$  contains nodes, then  $\mathcal{G}$  need not be a Voronoi diagram even if we can find a suitable  $(S, \sigma)$ . In Fig. 8, we illustrate two examples which both are bisector graphs for the same pair  $(S, \sigma)$  but only Fig. 8b is a Voronoi diagram. Observe that Fig. 8a does not contain the face  $f$  in the center of Fig. 8b.

Would it help to get additional information on the input graph  $\mathcal{G}$  or on  $\mathcal{VD}_\sigma(S)$  for an output  $(S, \sigma)$  obtained by our reconstruction algorithm? If it is known that  $\mathcal{VD}_\sigma(S)$  has no disconnected Voronoi region then  $\mathcal{G}$  is guaranteed to be identical to  $\mathcal{VD}_\sigma(S)$ . Similarly, if we are told that  $\mathcal{G}$  forms the Voronoi diagram of some unknown weighted input then we also have a good chance that an appropriate input can be reconstructed: If our reconstruction algorithm returns a unique solution  $(S, \sigma)$  such that  $\mathcal{G}$  is a bisector graph for  $(S, \sigma)$  then  $\mathcal{G}$  is also guaranteed to match  $\mathcal{VD}_\sigma(S)$ . In the unlikely case that we get two solutions for which  $\mathcal{G}$  forms a bisector graph then it is only clear that  $\mathcal{G}$  matches the Voronoi diagram of at least one of the solutions. Unfortunately, in this case there seems no simple way to pick the appropriate solution.

A canonical way to verify whether  $\mathcal{G}$  matches the Voronoi diagram  $\mathcal{VD}_\sigma(S)$  is to compute  $\mathcal{VD}_\sigma(S)$  using the approach by Aurenhammer and Edelsbrunner [2]. Their algorithm is worst-case optimal and runs in  $\mathcal{O}(n^2)$  time and space. Of course, this is a waste of time if the combinatorial complexity of  $\mathcal{G}$  is sub-quadratic. Alternatively, one may use the strategies presented by Har-Peled and Raichel [3] or Held and de Lorenzo [4], which allow to compute  $\mathcal{VD}_\sigma(S)$  in expected  $\mathcal{O}(n \log^3 n)$  or  $\mathcal{O}(n \log^4 n)$  time, respectively, under the assumption that the corresponding weights are sampled from some random distribution. A final comparison between the diagram computed and  $\mathcal{G}$  yields the decision sought: We find a common node in  $\mathcal{G}$  and  $\mathcal{VD}_\sigma(S)$  and apply a breath-first traversal to compare all nodes and arcs. Therefore, the comparison can be carried out in  $\mathcal{O}(m)$  time.

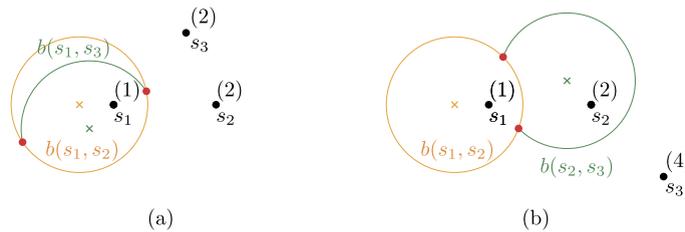
### 5. Discussion

We present a novel approach for recognizing whether a given planar circular-arc graph  $\mathcal{G}$  is a weighted bisector graph, and, provided that this is the case, for reconstructing the respective input sites. The Möbius transformation is central to our approach, as it allows us to generate a solution set, or confirm that  $\mathcal{G}$  is no weighted bisector graph whenever no such set exists.

It is obvious that we need Condition (2) in the definition of a bisector graph (at the end of Section 1.2) in order to permit a meaningful recognition. Without Condition (2), every circular-arc graph  $\mathcal{G}$  would constitute a bisector graph, and we could simply obtain a suitable pair of candidate sites for every edge  $e$  of  $\mathcal{G}$  by means of circular inversion across the supporting circle of  $e$ .

Fig. 9 shows two circular-arc graphs that both meet Condition (2) but violate Condition (3). (In Fig. 9a, all three edges have  $s_1$  as one of their defining sites, while  $s_2$  is one of the defining sites of all three edges in Fig. 9b.) If no further constraints are given then we can choose the position and weight of one of these three sites arbitrarily and then obtain the two other sites via circular inversions. E.g., for the setting shown in Fig. 9a, we could choose  $s_1$  and then obtain  $s_2$  and  $s_3$  via circular inversions of  $s_1$ , or choose  $s_2$  and then obtain  $s_1$  and then use  $s_1$  to obtain  $s_3$ . If, however, one of these sites is also subject to some other constraint imposed by  $\mathcal{G}$  that fixes its position and weight then all three sites would be determined (or no solution exists). Hence, our approach could be generalized to circular-arc graphs that violate Condition (3).

Whenever  $\mathcal{G}$  does not contain nodes, we are even able to determine whether  $\mathcal{G}$  is a multiplicatively weighted Voronoi diagram  $\mathcal{VD}_\sigma(S)$ , and to reconstruct  $(S, \sigma)$ . If a general graph  $\mathcal{G}$  has been recognized as a bisector graph of  $(S, \sigma)$  then the main difficulty of verifying that  $\mathcal{G}$  matches  $\mathcal{VD}_\sigma(S)$  is given by deciding whether all disconnected faces of  $\mathcal{VD}_\sigma(S)$  that do not contain their defining sites are also present in  $\mathcal{G}$ . It remains a question for future research to confirm that a bisector graph of  $(S, \sigma)$  matches  $\mathcal{VD}_\sigma(S)$  without explicitly computing  $\mathcal{VD}_\sigma(S)$ .



**Fig. 9.** Two circular-arc graphs that meet Condition (2) but violate Condition (3) in the definition of a bisector graph: The nodes (depicted in red) have two edges incident that lie on the same bisector circle  $b(s_1, s_2)$ .

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

No data was used for the research described in the article.

### Acknowledgements

This work was supported by the Austrian Science Fund (FWF): Grant P31013. We thank an anonymous reviewer for being persistent: The reviewer's repeated questions and remarks helped to reveal a mismatch between our algorithm and the original definition of "bisector graph".

### References

- [1] O. Aichholzer, F. Aurenhammer, D. Alberts, B. Gärtner, A novel type of skeleton for polygons, *J. Univers. Comput. Sci.* 1 (12) (1995) 752–761, [https://doi.org/10.1007/978-3-642-80350-5\\_65](https://doi.org/10.1007/978-3-642-80350-5_65).
- [2] F. Aurenhammer, H. Edelsbrunner, An optimal algorithm for constructing the weighted Voronoi diagram in the plane, *Pattern Recognit.* 17 (2) (1984) 251–257, [https://doi.org/10.1016/0031-3203\(84\)90064-5](https://doi.org/10.1016/0031-3203(84)90064-5).
- [3] S. Har-Peled, B. Raichel, On the complexity of randomly weighted multiplicative Voronoi diagrams, *Discrete Comput. Geom.* 53 (3) (2015) 547–568, <https://doi.org/10.1007/s00454-015-9675-0>.
- [4] M. Held, S. de Lorenzo, An efficient, practical algorithm and implementation for computing multiplicatively weighted Voronoi diagrams, in: *Proc. 28th Annu. Europ. Symp. Alg. (ESA'20)*, 2020, pp. 9:1–9:15.
- [5] B.N. Boots, Weighting thiesen polygons, *Econ. Geogr.* 56 (3) (1980) 248–259, <https://doi.org/10.2307/142716>.
- [6] G. Eder, M. Held, Weighted Voronoi diagrams in the maximum norm, *Int. J. Comput. Geom. Appl.* 29 (03) (2019) 239–250, <https://doi.org/10.1142/S0218195919500079>.
- [7] J.N. Portela, M.S. Alencar, Cellular coverage map as a Voronoi diagram, *J. Commun. Inf. Syst.* 23 (1) (2008) 22–31, <https://doi.org/10.14209/jcis.2008.3>.
- [8] M. Parter, D. Peleg, On the relations between SINR diagrams and Voronoi diagrams, in: *Proc. 14th Int. Conf. Ad-Hoc, Mobile, and Wireless Networks*, 2015, pp. 225–237.
- [9] A.R. David González G., Harri Hakula, J. Hämäläinen, Spatial mappings for planning and optimization of cellular networks, *IEEE/ACM Trans. Netw.* 26 (1) (2018) 175–188, <https://doi.org/10.1109/TNET.2017.2768561>.
- [10] P.F. Ash, E.D. Bolker, Recognizing Dirichlet tessellations, *Geom. Dedic.* 19 (2) (1985) 175–206, <https://doi.org/10.1007/BF00181470>.
- [11] D. Hartvigsen, Recognizing Voronoi diagrams with linear programming, *ORSA J. Comput.* 4 (4) (1992) 369–374, <https://doi.org/10.1287/ijoc.4.4.369>.
- [12] F. Aurenhammer, Recognizing polytopical cell complexes and constructing projection polyhedra, *J. Symb. Comput.* 3 (3) (1987) 249–255, [https://doi.org/10.1016/S0747-7171\(87\)80003-2](https://doi.org/10.1016/S0747-7171(87)80003-2).
- [13] T. Biedl, M. Held, S. Huber, Recognizing straight skeletons and Voronoi diagrams and reconstructing their input, in: *Proc. 10th Int. Sympos. Voronoi Diagrams in Sci. & Eng. (ISVD'13)*, 2013, pp. 37–46.
- [14] O. Aichholzer, T. Biedl, T. Hackl, M. Held, S. Huber, P. Palfrader, B. Vogtenhuber, Representing directed trees as straight skeletons, in: *Proc. 23rd Int. Symp. Graph Drawing & Network Visualization (GD 2015)*, 2015, pp. 335–347.
- [15] O. Aichholzer, H. Cheng, S.L. Devadoss, T. Hackl, S. Huber, B. Li, A. Risteski, What makes a tree a straight skeleton?, in: *Proc. 24th Canad. Conf. Comp. Geom. (CCCG'12)*, 2012, pp. 267–272.
- [16] G. Eder, M. Held, P. Palfrader, Recognizing geometric trees as positively weighted straight skeletons and reconstructing their input, *Int. J. Comput. Geom. Appl.* 29 (03) (2019) 251–267, <https://doi.org/10.1142/S0218195919500080>.
- [17] N. Altshiller-Court, *College Geometry: An Introduction to the Modern Geometry of the Triangle and the Circle*, Dover Publications, 2007.
- [18] H. Schwerdtfeger, *Geometry of Complex Numbers: Circle Geometry, Moebius Transformation, Non-Euclidean Geometry*, Dover Publications, 1980, ISBN:978-0486638300.
- [19] K.Q. Brown, *Geometric Transforms for Fast Geometric Algorithms*, Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1979.