A Wavefront-Like Strategy for Computing Multiplicatively Weighted Voronoi Diagrams

M. Held[1] and S. de Lorenzo[2]

1 **University of Salzburg**
`held@cs.sbg.ac.at`
2 **University of Salzburg**
`slorenzo@cs.sbg.ac.at`

Abstract

We study multiplicatively weighted Voronoi diagrams (MWVDs) of point sites in the Euclidean plane and present a wavefront-like approach for computing the MWVD of n points in near-optimal $\mathcal{O}(n^2 \log n)$ time and $\Theta(n^2)$ space. The key advantage of our algorithm is its simplicity. Furthermore, it can be extended to handle additive weights at no additional computational cost.

1 Introduction

Let S denote a finite set of n distinct weighted points in $\mathbb{R}^2$, so-called *sites*. A weight function $w\colon S \to \mathbb{R}^+$ assigns a strictly positive weight $w(s)$ to every site $s \in S$. It is common to regard the weighted Euclidean distance $d_w(p, s)$ from an arbitrary point p in $\mathbb{R}^2$ to a site $s \in S$ as the standard Euclidean distance $d(p, s)$ from p to s divided by the weight of s:

$$d_w(p, s) := \frac{1}{w(s)} \cdot d(p, s).$$

For $s \in S$, the *(weighted) Voronoi region* $\mathcal{VR}_w(s, S)$ of s relative to S is the set of all points of the plane that are not farther to s than to any other site s' in S, that is

$$\mathcal{VR}_w(s, S) := \{p \in \mathbb{R}^2 : d_w(p, s) \leq d_w(p, s') \text{ for all } s' \in S \text{ with } s \neq s'\}.$$

The multiplicatively weighted Voronoi diagram (MWVD), $\mathcal{VD}_w(S)$, of S is simply defined as

$$\mathcal{VD}_w(S) := \bigcup_{s \in S} \partial\, \mathcal{VR}_w(s, S).$$

An example of a MWVD is shown in Figure 1.

A connected component of a Voronoi region is called a *face*. Voronoi region. For two distinct sites s, s' of S, the *bisector* $b(s, s')$ of s, s' models the set of points of the plane that are at the same weighted distance from s and s'. Hence, a non-empty intersection of two Voronoi regions is a subset of the bisector of the two defining sites. Following common terminology, a connected component of such a set is called a *(Voronoi) edge* of $\mathcal{VD}_w(S)$. An endpoint of an edge is called a *(Voronoi) node*. It is known that the bisector between two unequally weighted sites forms a circle[1]. Hence, the Voronoi edges of $\mathcal{VD}_w(S)$ are given by straight-line segments and circular arcs. In contrast to the standard Voronoi diagram, a MWVD may include a quadratic number of Voronoi nodes, edges, and faces [2].

In the sequel we present work in progress on computing MWVDs. Our current algorithm operates entirely in the plane and runs in $\mathcal{O}(n^2 \log n)$ time and $\Theta(n^2)$ space. It is based on a

[1] Apollonius of Perga defined a circle as a set of points that have a specific distance ratio to two foci.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

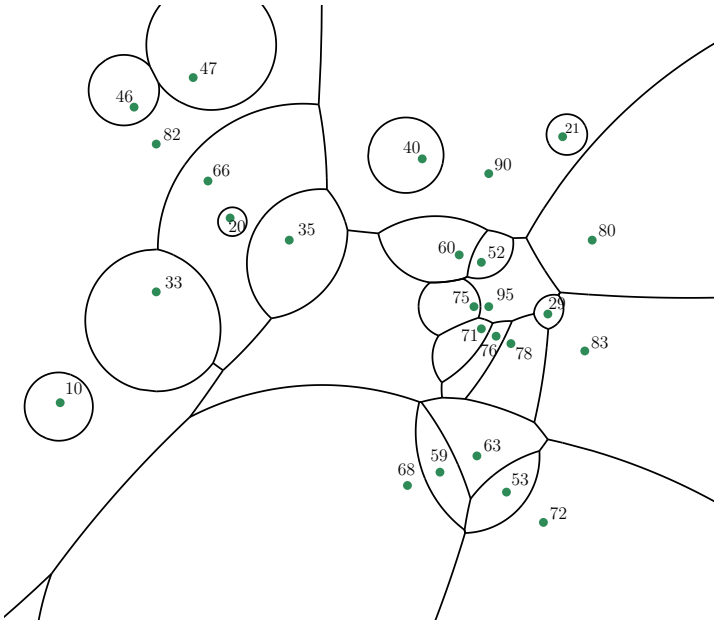

Figure 1 The MWVD (shown in black) of a set of input points (in green). The numbers next to the points indicate their weights.

wavefront-like expansion of weighted circles. As time progresses the wavefront $\mathcal{WF}(t)$ covers a growing portion of $\mathbb{R}^2$. Certain events mark topological changes of $\mathcal{WF}(t)$ and aid us in finding the individual Voronoi nodes. Although it is no wavefront propagation in the strict meaning of the word, we will (for the sake of simplicity) omit the qualifier "like" from here on. Our approach can be extended to handle both additive and multiplicative weights at no additional cost. (Details are omitted due to lack of space.) We developed a prototype implementation of our algorithm in C++ which uses standard IEEE 754 floating-point arithmetic.

2 Related Work

MWVDs were initially studied by Boots [3] in terms of market area analysis. In 1984, Aurenhammer and Edelsbrunner [2] presented an optimal algorithm for constructing the MWVD of a set of n points in $\mathbb{R}^2$ in $\mathcal{O}(n^2)$ time and space. They define spheres on the bisector circles and convert them into half-planes using a spherical inversion. We are not aware of an implementation of their algorithm, though. Later Aurenhammer uses divide&conquer to obtain an $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space algorithm for the one-dimensional weighted Voronoi diagram [1], where all weighted input points lie on a line. Har-Peled and Raichel [4] show that MWVDs have a slightly super-linear expected combinatorial complexity if the weights are chosen randomly. Their result provides the motivation for working on an algorithm whose running time is output-sensitive.

Vyatkina and Barequet [5] present a wavefront-like strategy to compute the weighted Voronoi diagram of a set of lines in the plane in $\mathcal{O}(n^2 \log n)$ time. The Voronoi nodes are computed based on edge and break-through events. An edge event takes place when an active arc vanishes. A break-through event happens whenever a new wavefront arc appears.

3 Algorithm

For the sake of descriptional simplicity we assume that no point in $\mathbb{R}^2$ has the same weighted distance to more than three input sites of S. For time $t \in \mathbb{R}_0^+$, each site $s \in S$ is associated with an expanding *offset circle* $o(s,t)$ which is centered at s and whose radius equals $t \cdot w(s)$. We find it convenient to regard $o(s,t)$ as a function of either time or distance since at time t every point on $o(s,t)$ is at Euclidean distance $t \cdot w(s)$ from s, i.e., at weighted distance t.

The wavefront $\mathcal{WF}(t)$ at time t is the set of all points p of the plane whose weighted distance from S equals t: We have $p \in \mathcal{WF}(t)$ if and only if $\min_{s \in S} d_w(p,s) = t$. The wavefront is formed by parts of offset circles which we will refer to as *wavefront arcs*. Every wavefront arc starts and ends at a moving *wavefront vertex*, i.e., a specific intersection point with another offset circle. These vertices will trace out the MWVD, see Figure 2.

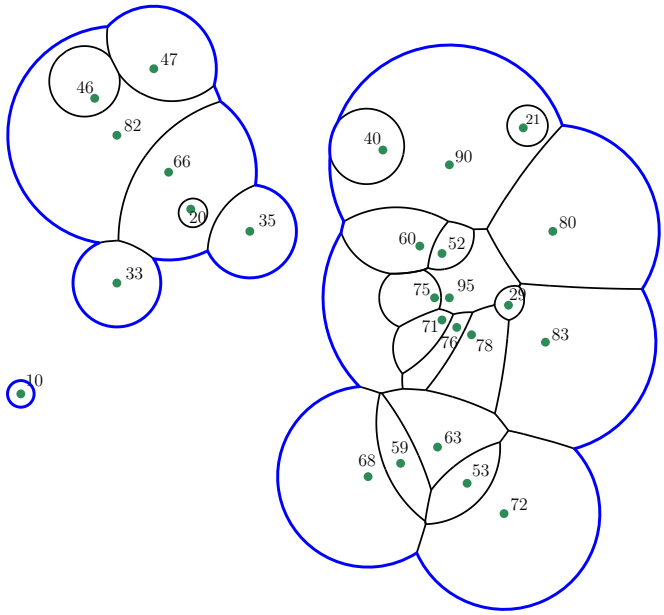

Figure 2 A wavefront $\mathcal{WF}(t)$ (in blue) and the MWVD traced out till some time t for the setting of Figure 1.

We now consider two sites $s_1, s_2 \in S$, with $w(s_1) < w(s_2)$, and assume that $o(s_1,t)$ and $o(s_2,t)$ intersect in the points i_1 and i_2. These two *moving intersection points* trace out $b(s_1,s_2)$. (Of course, the moving intersection point i_1 depends on time t but we simply write i_1 instead of $i_1(t)$.) Since $w(s_1) < w(s_2)$, it is easy to see that the arc of $o(s_1,t)$ which is inside $o(s_2,t)$ will not belong to $\mathcal{WF}(t')$ for any $t' > t$. We refer to such an arc of an offset circle as *inactive*. All other arcs in the arrangement of all offset circles are called *active*, see Figure 3. Thus, it is necessary (but not sufficient) for an arc of an offset circle to be active for all times t' with $0 \leq t' < t$ if it is active at time t.

We now describe an event-handling scheme that allows to trace out the Voronoi diagram by simulating the expansion of all active arcs. During the wavefront propagation process, *collision events* mark the initial contact of two offset circles, *domination events* happen as soon as the offset circle of a higher-weighted site fully contains the offset circle of a lower-weighted one, and *edge events* as well as *break-through events* take place whenever an active arc vanishes or appears. These events capture topological changes of the wavefront and determine the corresponding Voronoi nodes of $\mathcal{VD}_w(S)$. The latter two events are triggered

Figure 3 The arrangement of all active arcs that corresponds to the wavefront depicted in Figure 2.

whenever one or more active arcs vanish, i.e., shrink to zero length.

Every site s keeps track of the active arcs of its expanding offset circle $o(s,t)$ by storing them in a self-balancing binary search tree $\mathcal{T}(s)$ that is updated whenever events occur. It maintains these arcs in sorted angular order as they appear when $o(s,t)$ is traversed counter-clockwise.

We start with computing the collision times for every pair of offset circles and insert them into a priority queue $\mathcal{Q}$. The initial wavefront $\mathcal{WF}(t)$, for $t > 0$ small enough, contains the full offset circles of all sites, and every offset circle forms one active arc. Every active arc is also marked to belong to $\mathcal{WF}(t)$.

As soon as the initialization phase is completed, the events are successively popped from $\mathcal{Q}$. Let e be the current event at time t_e. Note that the number and order of the active arcs along a specific offset circle cannot change between two consecutive events, but their extents, i.e., the portions of the offset circle which they occupy, may change. Hence it is important to recompute the positions of the moving intersection points on the fly whenever we perform a search in one of our search trees at time t_e.

Collision event: If e is a collision event then two offset circles $o(s_1,t_e)$ and $o(s_2,t_e)$ meet at a single point q for the first time; see Figure 4. We search $\mathcal{T}(s_1)$ and $\mathcal{T}(s_2)$, and determine the arcs a_1 and a_2 which contain q. If either of them is inactive then this event requires no further processing. Otherwise we create two moving intersections i'_{12} and i''_{12} at q and we split both a_1 and a_2 at q. W.l.o.g., $w(s_1) < w(s_2)$. As time progresses, i'_{12} and i''_{12} limit a new active arc on $o(s_2,t)$ and an inactive arc on $o(s_1,t)$. (If $w(s_1) = w(s_2)$ then both new arcs are inactive.)

Domination Event: If e is a domination event then the offset circle $o(s_1,t_e)$ fully contains the offset circle $o(s_3,t_e)$ for the first time at time t_e, with $w(s_3) < w(s_1)$; see Figure 5. That

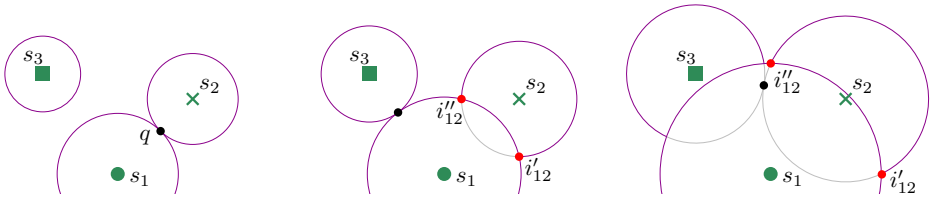

Figure 4 The collision events (depicted by black dots) of the offset circles that correspond to the three input sites s_1, s_2, and s_3, where $w(s_1) := 5$, $w(s_2) := 6$, and $w(s_3) := 8$. The active arcs of the offset circles are drawn in magenta whereas the inactive arcs are shown in gray. The red dots indicate the moving intersection points i'_{12} and i''_{12}.

is, the two offset circles touch at a point q. Let a_1 and a_3 be the corresponding arcs that contain q. If one of them is inactive then we ignore this event. Otherwise, all arcs of $o(s_3, t_e)$ become inactive and a_1 is merged with its neighboring arcs.

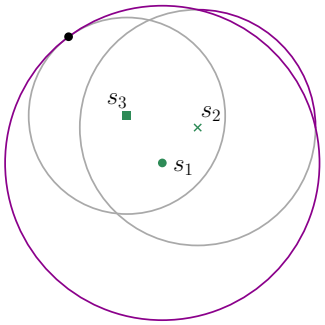

Figure 5 A domination event that involves s_1 and s_3 takes place.

Edge Event: An edge event occurs at time t_e if at least one active arc along an offset circle shrinks to a point q, i.e., to zero length; see Figure 6. If an entire circular-arc triangle shrinks to q at time t_e then all active arcs involved can be removed from their corresponding offset circles. Otherwise, a single active arc a_1 on one of the two higher weighted offset circles $o(s_1, t_e)$ just disappeared. The two other sites s_2 and s_3, where s_3 is w.l.o.g. the site with the lowest weight, whose offset circles cause a_1 to vanish can be derived from the moving intersection points i_{12} and i_{13} that limit a_1. We remove a_1 from $o(s_1, t_e)$ and add a new active arc which is bound by i_{12} and i_{23} to $o(s_2, t_e)$. Additionally, the moving intersection point i_{13} that bounds an active arcs along $o(s_3, t_e)$ needs to be replaced by i_{23}.

Break-Through Event: A break-through event is a special kind of edge event and can be treated similarly. It also occurs if an active arc a_1 arc on $o(s_2, t_e)$ shrinks to a point q. Let, again, s_1 and s_3 be the other two sites that participate in this event, where s_3 is associated with the lowest weight; see Figure 7. The arc a_1 is deleted from $o(s_2, t_e)$ and a new active arc a'_1 spawns between the corresponding moving intersection points i_{12} and i_{13} on $o(s_1, t_e)$, and i_{23} that bounds a neighboring active arc along $o(s_3, t_e)$ is replaced by i_{13}.

Common to all these events is the necessity to compute and store a future edge event for every newly created active arc. Furthermore, every inactive arc is deleted from its search data structure, while every new active arc is inserted into the search data structure of its site.

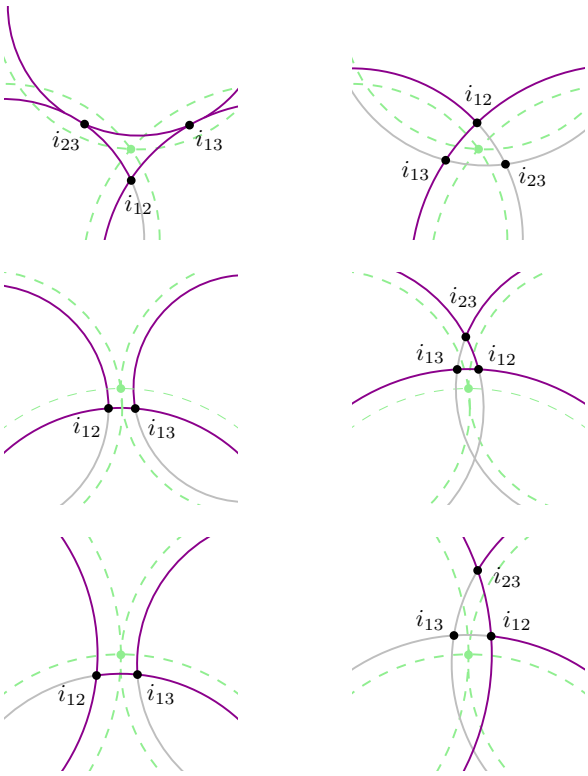

Figure 6 In the top-most subfigure the configuration before and after the collapse of an entire circular-arc triangle is displayed. The remaining subfigures illustrate the collapse of a single active arc on the offset circle of the highest-weighted (medium-weighted, resp.) site.

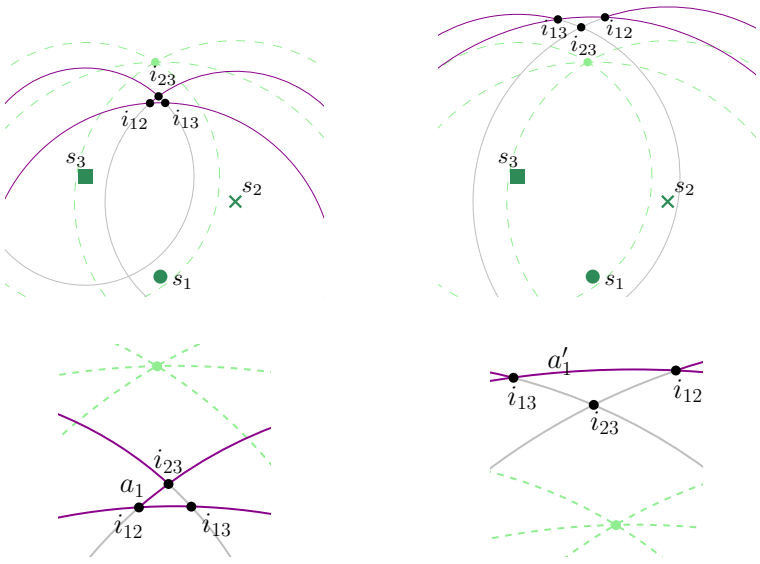

Figure 7 An active arc a_1 disappears and a new active wavefront arc a_1' spawns between the moving intersection points i_{12} and i_{13} at a break-through event.

We process events until the priority queue $\mathcal{Q}$ is empty. If the largest weight is associated with a strict subset $S_{\max}$ of S then $\mathcal{Q}$ will be empty once the wavefront contains only arcs of offset circles of sites of $S_{\max}$. If all sites have identical weight then $\mathcal{VD}_w(S)$ equals the standard Voronoi diagram and $\mathcal{Q}$ will be empty once the wavefront contains only arcs of offset circles of sites which lie on the convex hull of S.

4 Analysis

All topological changes of the wavefront are properly detected by our algorithm, as they coincide with the collapse of at least one active arc of the wavefront. It remains to determine an upper bound on the number of events that may take place during the wavefront propagation.

▶ **Lemma 4.1.** *During the wavefront propagation $\Theta(n^2)$ many collision and $\mathcal{O}(n^2)$ many domination events are computed.*

Proof. Recall that every pair of input sites corresponds to at most one collision and at most one domination event, with all collision events being computed a priori. ◀

▶ **Lemma 4.2.** *During the wavefront propagation $\mathcal{O}(n^2)$ many break-through events occur.*

Proof. Let s_1, s_2 and s_3 be the sites whose expanding offset circles $o(s_1, t)$, $o(s_2, t)$ and $o(s_3, t)$ are involved in a break-through event at time t_e. These offset circles define the three moving intersection points i_{12}, i_{13} and i_{23}, where i_{23} is shared by active arcs of $o(s_2, t)$ and $o(s_3, t)$ for $t < t_e$; recall Figure 7. At the time of the event, the offset circle $o(s_1, t_e)$ passes through i_{23} and this moving intersection point will be contained inside of $o(s_1, t)$ for all $t > t_e$. Hence, for $t > t_e$, no active arc of $o(s_2, t)$ can share a common vertex with an active arc of $o(s_3, t)$. (Note that otherwise the MWVD of $\{s_1, s_2, s_3\}$ would potentially include multiply-connected Voronoi regions.) This implies that a break-through event can occur at most once for each pair of input sites. ◀

▶ **Lemma 4.3.** *During the wavefront propagation $\mathcal{O}(n^2)$ many edge events occur.*

Proof. At every collision and domination event a constant number of new active arcs is generated, and every break-through event results in exactly one new active arc, resulting in a total of $\mathcal{O}(n^2)$ new active arcs during the entire run of the wavefront propagation. Every edge event either reduces the number of active arcs by at least one or, if this number stays constant, then it is coupled to exactly one of the at most quadratically many break-through events, recall the right part of Figure 6. ◀

Summarizing, $\Theta(n^2)$ events take place during the wavefront propagation. Each of these events consumes up to $\mathcal{O}(\log n)$ time, since every event requires a constant number of lookups, insertions, and/or deletions in a self-balancing binary search tree of size $\mathcal{O}(n)$ or in a priority queue of size $\Theta(n^2)$. Thus, we get an overall runtime of $\mathcal{O}(n^2 \log n)$. Additionally, this algorithm requires $\Theta(n^2)$ memory because our current approach computes all quadratically many collision events a priori. Avoiding this computational burden is work in progress.

Acknowledgments Work supported by Austrian Science Fund (FWF): Grant P31013-N31.

References

1 Franz Aurenhammer. The One-Dimensional Weighted Voronoi Diagram. *Information Processing Letters*, 22(3):119–123, 1986. doi:10.1016/0020-0190(86)90055-4.

2 Franz Aurenhammer and Herbert Edelsbrunner. An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane. *Pattern Recognition*, 17(2):251–257, 1984. doi:10.1016/0031-3203(84)90064-5.

3 Barry N Boots. Weighting Thiessen Polygons. *Economic Geography*, 56(3):248–259, 1980. doi:10.2307/142716.

4 Sariel Har-Peled and Benjamin Raichel. On the Complexity of Randomly Weighted Multiplicative Voronoi Diagrams. *Discrete & Computational Geometry*, 53(3):547–568, 2015. doi:10.1007/s00454-015-9675-0.

5 Kira Vyatkina and Gill Barequet. On Multiplicatively Weighted Voronoi Diagrams for Lines in the Plane. *Trans. Computational Science*, 13:44–71, 2011. doi:10.1007/978-3-642-22619-9_3.