Separating a Voronoi Diagram via Local Search

Vijay V. S. P. Bhattiprolu[1] **and Sariel Har-Peled**[*2]

1 **School of Computer Science, Carnegie Mellon University, Pittsburgh, USA**
vpb@cs.cmu.edu

2 **Department of Computer Science, University of Illinois, Urbana, USA**
sariel@illinois.edu

Abstract

Given a set P of n points in $\mathbb{R}^d$, we show how to insert a set Z of $O\left(n^{1-1/d}\right)$ additional points, such that P can be broken into two sets P_1 and P_2, of roughly equal size, such that in the Voronoi diagram $\mathcal{V}(P \cup Z)$, the cells of P_1 do not touch the cells of P_2; that is, Z separates P_1 from P_2 in the Voronoi diagram (and also in the dual Delaunay triangulation). In addition, given such a partition (P_1, P_2) of P, we present an approximation algorithm to compute a minimum size separator realizing this partition. We also present a simple local search algorithm that is a PTAS for approximating the optimal Voronoi partition.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, I.1.2 Algorithms, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Separators, Local search, Approximation, Voronoi diagrams, Delaunay triangulation, Meshing, Geometric hitting set

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.18

1 Introduction

Many algorithms work by partitioning the input into a small number of pieces, of roughly equal size, with little interaction between the different pieces, and then recurse on these pieces. One natural way to compute such partitions for graphs is via the usage of separators. A (vertex) separator of a graph $G = (V, E)$, informally, is a "small" set $Z \subseteq V$ whose removal breaks the graph into two (or more) subgraphs, each of which is of size at most n/c, where c is some constant strictly larger than one. As a concrete example, any tree with n vertices has a single vertex, which can be computed in linear time, such that its removal breaks the tree into subtrees, each with at most $n/2$ vertices.

Separators in planar graphs. Ungar [34] showed that a planar graph with n vertices contains a separator of size $O\left(\sqrt{n \log n}\right)$. This was later improved by Lipton and Tarjan [24] to $O(\sqrt{n})$, and they also provided an algorithm to compute the separator in linear time. Specifically, there exists a separator of size $O(\sqrt{n})$ such that its removal partitions the graph into two disjoint subgraphs each containing at most $2n/3$ vertices (each of these subgraphs is not necessarily connected).

There has been a substantial amount of work on planar separators in the last four decades, and they are widely used in data-structures and algorithms for planar graphs, including (i) shortest paths [15], (ii) distance oracles [33], (iii) max flow [13], and (iv) approximation

* Work on this paper by the second author was partially supported by NSF AF awards CCF-0915984, CCF-1421231, and CCF-1217462.

32nd International Symposium on Computational Geometry (SoCG 2016).
Editors: Sándor Fekete and Anna Lubiw; Article No. 18; pp. 18:1–18:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

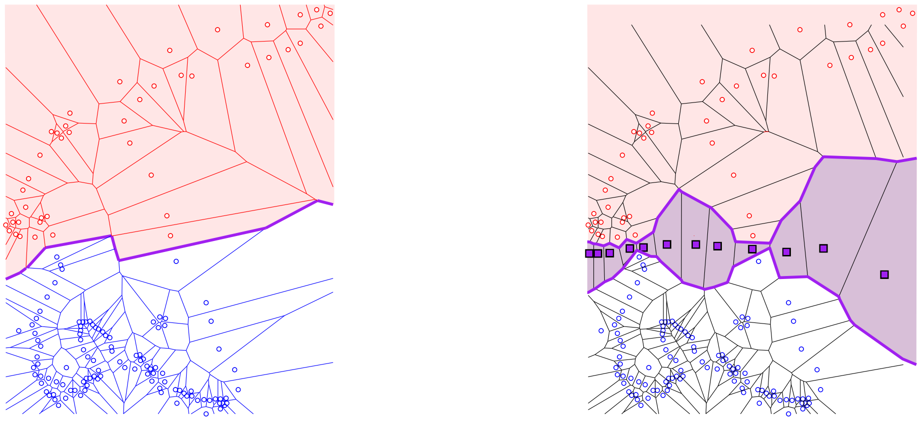


Figure 1 On the left a Voronoi partition, and on the right, a separator realizing it.

algorithms for TSP [22]. This list is a far cry from being exhaustive, and is a somewhat arbitrary selection of some recent results on the topic.

Geometric separators. Any planar graph can be realized as a set of interior disjoint disks, where a pair of disks touch each other, if and only if the corresponding vertices have an edge between them. This is the *circle packing theorem* [31], also known as the Koebe-Andreev-Thurston theorem [23].

Surprisingly, the existence of a planar separator is an easy consequence of the circle packing theorem. This was proved by Miller *et al.* [26] (see also [19]). Among other things, Miller *et al.* showed that given a set of n balls in $\mathbb{R}^d$, such that no point is covered more than k times, the intersection graph of the balls has a separator of size $O\left(k^{1/d}n^{1-1/d}\right)$. This implies that the k-nearest neighbor graph of a set of points in $\mathbb{R}^d$, has a small separator [26, 19]. Various extensions of this technique were described by Smith and Wormald [32].

Other separators. Small separators are known to exist for many other families of graphs. These include graphs (i) with bounded tree width [8], (ii) with bounded genus [16], (iii) that are minor free [1], and (iv) grids. Furthermore, graphs with hereditary sublinear separators have *polynomial expansion* [11], and vice versa – graphs with polynomial expansion have sublinear separators [29].

Voronoi separators. In this paper, we are interested in geometric separation in a Voronoi diagram [3]. Specifically, given a set P of points in $\mathbb{R}^d$, we are interested in inserting a small set of new points Z, such that there is a balanced partition of P into two sets P_1, P_2, such that no cell of P_1 touches a cell of P_2 in the Voronoi diagram $\mathcal{V}(P \cup Z)$. Note, that such a set Z also separates P_1 and P_2 in the Delaunay triangulation of $P \cup Z$.

Using Voronoi separators. One of the motivations of Lipton and Tarjan [24, 25] was implementing divide and conquer algorithms on graphs. For example, *generalized nested dissection* – solving a system of linear equations rising out of numerical simulations done over planar meshes. Thus, Voronoi separators (which in the dual are *Delaunay separators*) provide a way to breakup Delaunay meshes. Unlike their planar graph counterpart, Voronoi separators also exist in higher dimensions.

Specifically, some meshing algorithms rely on computing a Delaunay triangulation of geometric models to get good triangulations that describe solid bodies. Such meshes in turn are fed into numerical solvers to simulate various physical processes. To get good triangulations, one performs a Delaunay refinement that involves inserting new points into the triangulations, to guarantee that the resulting elements are well behaved. Since the underlying geometric models can be quite complicated and these refinement processes can be computationally intensive, it is natural to try and break up the data in a balanced way, and Voronoi separators provide one way to do so. In particular, small Voronoi separators provide a way to break up a point set in such a way that there is limited interaction between two pieces of the data.

Geometric hitting set. Given a set of objects in $\mathbb{R}^d$, the problem of finding a small number of points that stab all the objects is an instance of geometric hitting set. There is quite a bit of research on this problem. In particular, the problem is NP-HARD for almost any natural instance, but a polynomial time $(1+\varepsilon)$-approximation algorithm is known for the case of balls in $\mathbb{R}^d$ [9], where one is allowed to place the stabbing points anywhere. The discrete variant of this problem, where there is a set of allowable locations to place the stabbing points, seems to be significantly harder and only weaker results are known [20]. See Mustafa *et al.* [27] for a QPTAS for the case of disks and points, and Har-Peled and Quanrud [21] for PTAS for shallow fat objects and matching hardness results.

One of the more interesting versions of the geometric hitting set problem, is the art gallery problem, where one is given a simple polygon in the plane, and one has to select a set of points (inside or on the boundary of the polygon) that "see" the whole polygon. While much research has gone into variants of this problem [30], nothing is known as far as an approximation algorithm (for the general problem). The difficulty arises from the underlying set system being infinite, see [12] for some efforts in better understanding this problem.

Geometric local search. Relatively little is known regarding local search methods for geometric approximation problems. Arya *et al.* [2] gave a local search method for approximating k-median clustering by a constant factor, and this was recently simplified by Gupta and Tangwongsan [17]. Mustafa and Ray [28] gave a local search algorithm for the discrete hitting set problem over pseudo disks and r-admissible regions in the plane, which yields a PTAS. Chan and Har-Peled [10] gave a local search PTAS for the independent set problem over fat objects, and for pseudodisks in the plane. Both works use separators in proving the quality of approximation.

More recently, Bandyapadhyay and Varadarajan [4] have employed some of the results in this paper to give a bi-criteria local search PTAS for variants of the k-means problem.

1.1 Our results

In this paper we give algorithms for the following:

(A) Computing a small Voronoi separator. Given a set P of n points in $\mathbb{R}^d$, we show how to compute, in expected linear time, a balanced Voronoi separator of size $O\big(n^{1-1/d}\big)$. This is described in Section 3. The existence of such a separator was not known before, and our proof is relatively simple and elegant.

(B) Exact algorithm for computing the smallest Voronoi separator realizing a given partition. In Section 4, given a partition (P_1, P_2) of a point set P in $\mathbb{R}^d$, we describe an algorithm that computes a minimum size Voronoi separator realizing this separation. The running time of the algorithm is $n^{O\left(d^2 \mathsf{b}\right)}$, where b is the cardinality of the optimal separating sets.

(C) Constant approximation algorithm for the smallest Voronoi separator realizing a given partition. In Section 5, we describe how to compute a constant factor approximation to the size of an optimal Voronoi separator for a given partition of a set in $\mathbb{R}^d$. This is the natural extension of the greedy algorithm for geometric hitting set of balls, except that in this case, the set of balls is infinite and is encoded implicitly, which somewhat complicates things.

(D) A PTAS for the smallest Voronoi separator realizing a given partition. We present a polynomial time approximation scheme to compute a Voronoi separator, realizing a given partition, whose size is a $(1+\varepsilon)$-approximation to the size of an optimal Voronoi separator for a given partition of a set in $\mathbb{R}^d$. The running time is $n^{O(1/\varepsilon^d)}$. See the full version of the paper for the full details [7], which are omitted in this version.
Interestingly, the new algorithm provides a PTAS for the geometric hitting set problem (for balls), that unlike previous approaches that worked top-down [9, 14], works more in a bottom-up approach. Note, that since our set of balls that needs to be pierced is infinite, and is defined implicitly, it is not obvious a priori how to use the previous algorithms in this case.

Sketch of algorithm. The new algorithm works by first computing a "dirty" constant approximation hitting set using a greedy approach (this is relatively standard). Somewhat oversimplifying, the algorithm next clusters this large hitting set into tight clusters of size $k = O(1/\varepsilon^d)$ each. It then replaces each such cluster of the weak hitting set, by the optimal hitting set that can pierce the same set of balls, computed by using the exact algorithm – which is "fast" since the number of piercing points is at most $O(1/\varepsilon^d)$. In the end of this process the resulting set of points is the desired hitting set. Namely, the new approximation algorithm reduces the given geometric hitting set instance, into $O(m/k)$ smaller instances where m is the size of the overall optimal hitting set and each of the smaller instances has an optimal hitting set of size $O(k)$.
For the analysis of this algorithm, we need a strengthened version of the separator theorem. See Theorem 23 for details.

(E) Local search PTAS for Voronoi partition and continuous geometric hitting set problems. An interesting consequence of the new bottom-up PTAS, is that it leads to a simple local search algorithm for geometric hitting set problems for fat objects. Specifically, in Section 7, we show that the algorithm starts with any hitting set (of the given objects) and continues to make local improvements via exchanges of size at most $O\left(1/\varepsilon^d\right)$, until no such improvement is possible, yielding a PTAS. The analysis of the local search algorithm is subtle and involves simultaneously clustering the locally optimal solution, and the optimal solution, and matching these clusters to each other.

Relation to known results. The separator result is similar in spirit (but not in details) to the work of Miller *et al.* [26] and Chan [9] on a separator for a k-ply set of balls – the main difference being that Voronoi cells behave differently than balls do. The bottom-up PTAS approach seems to be new, and should be applicable to other problems. Having said that, it seems like the top-down approaches [9, 14] potentially can be modified to work in this case, but the low level details seem to be significantly more complicated, and the difficulty in making them work was the main motivation for developing the new approach. The basic idea of using separators in analyzing local search algorithms appear in the work of Mustafa and Ray [28] and Chan and Har-Peled [10].

2 Preliminaries

For a point set $P \subseteq \mathbb{R}^d$, the *Voronoi diagram* of P, denoted by $\mathcal{V}(P)$ is the partition of space into convex cells, where the *Voronoi cell* of $\mathsf{p} \in P$ is

$$\mathcal{C}_P(\mathsf{p}) = \left\{ \mathsf{q} \in \mathbb{R}^d \;\middle|\; \|\mathsf{q} - \mathsf{p}\| \leq \mathsf{d}(\mathsf{q}, P) \right\},$$

where $\mathsf{d}(\mathsf{q}, P) = \min_{\mathsf{s} \in P} \|\mathsf{q} - \mathsf{s}\|$ is the distance of q to the set P. Voronoi diagrams are a staple topic in Computational Geometry, see [6], and we include the definitions here for the sake of completeness.

In the plane, the Voronoi diagram has linear descriptive complexity. Here, the complexity refers to the length of encoding a set as a semialgebraic set [5]. Specifically, the diagram can be broken into linear number of cells (i.e., triangles in this case), where each cell can be described by a constant number of algebraic inequalities.

For a point set P, and points $\mathsf{p}, \mathsf{q} \in P$, the geometric loci of all points in $\mathbb{R}^d$ that have both p and q as nearest neighbor, is the *bisector* of p and q – it is denoted by $\beta_{\mathsf{p},\mathsf{q}} = \left\{ \mathsf{s} \in \mathbb{R}^d \;\middle|\; \|\mathsf{s} - \mathsf{p}\| = \|\mathsf{s} - \mathsf{q}\| = \mathsf{d}(\mathsf{s}, P) \right\}$. A point $\mathsf{s} \in \beta_{\mathsf{p},\mathsf{q}}$ is the center of a ball whose interior does not contain any point of P and that has p and q on its boundary. The set of all such balls induced by $\beta_{\mathsf{p},\mathsf{q}}$ is the *pencil* of p and q, denoted by $\mathrm{pencil}(\mathsf{p}, \mathsf{q})$.

▶ **Definition 1.** Let P be a set of points in $\mathbb{R}^d$, and P_1 and P_2 be two disjoint subsets of P. The sets P_1 and P_2 are *Voronoi separated* in P if for all $\mathsf{p}_1 \in P_1$ and $\mathsf{p}_2 \in P_2$, we have that their Voronoi cells are disjoint; that is, $\mathcal{C}_P(\mathsf{p}_1) \cap \mathcal{C}_P(\mathsf{p}_2) = \emptyset$.

▶ **Definition 2.** For a set P, a *partition* of P is a pair of sets (P_1, P_2), such that $P_1 \subseteq P$, and $P_2 = P \setminus P_1$. A set Z is a *Voronoi separator* for a partition (P_1, P_2) of $P \subseteq \mathbb{R}^d$, if P_1 and P_2 are Voronoi separated in $P \cup Z$; that is, the Voronoi cells of P_1 in $\mathcal{V}(P \cup Z)$ do not intersect the Voronoi cells of P_2. We will refer to the points of the separator Z as *guards*.

See Figure 1 for an example of the above definitions.

▶ **Definition 3.** For a ball b, its *cover number* is the minimum number of (closed) balls of half the radius that are needed to cover it. The *doubling constant* of a metric space is the maximum cover number over all possible balls. Let c^d_{dbl} be the doubling constant for $\mathbb{R}^d$.

The constant c^d_{dbl} is exponential in d, and $c^d_{\mathrm{dbl}} \leq \left\lceil 2\sqrt{d} \right\rceil^d$ – indeed, cover a ball (say, of unit radius) by a grid with sidelength $1/\sqrt{d}$, and observe that each grid cell has diameter 1, and as such can be covered by a ball of radius $1/2$.

▶ **Definition 4.** For a closed set $\mathsf{X} \subseteq \mathbb{R}^d$, and a point $\mathsf{p} \in \mathbb{R}^d$, the *projection* of p into X is the closest point in X to p. We denote the projected point by $\mathsf{nn}(\mathsf{p}, \mathsf{X})$.

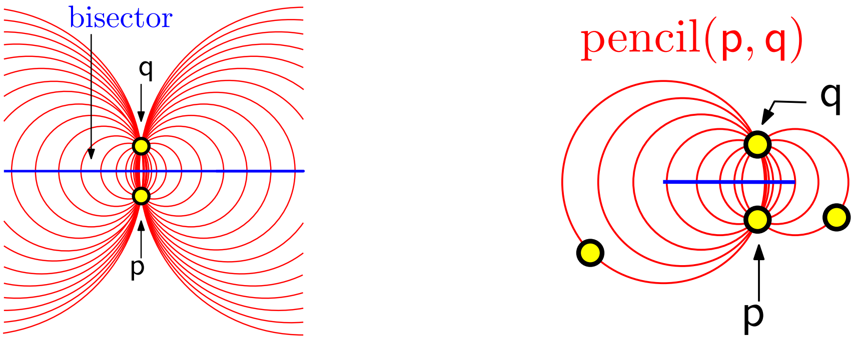
3 Computing a small Voronoi separator

3.1 Preliminaries and how to guard a sphere

Given a set P of n points in $\mathbb{R}^d$, we show how to compute a balanced Voronoi separator for P of size $O\left(n^{1-1/d}\right)$.

▶ **Definition 5.** A set $\mathsf{Y} \subseteq \mathsf{X} \subseteq \mathbb{R}^d$ is *ℓ-dense* in X, if for any point $\mathsf{p} \in \mathsf{X}$, there exists a point $\mathsf{s} \in \mathsf{Y}$, such that $\|\mathsf{p} - \mathsf{s}\| \leq \ell$.

▶ **Lemma 6** (proof in [7]). *Consider an arbitrary sphere $\mathbb{S}$, and a point $\mathsf{p} \in \mathbb{R}^d \setminus \mathbb{S}$. Then one can compute, in constant time, a set of points $Q \subseteq \mathbb{S}$, such that the Voronoi cell $\mathcal{C}_{Q \cup \{\mathsf{p}\}}(\mathsf{p})$ does not intersect $\mathbb{S}$, and $|Q| = O(1)$. We denote the set Q by* $\mathrm{blockerSet}(\mathsf{p}, \mathbb{S})$.



(a) **(b)**

Figure 2 (a) The unbounded bisector induced by p and q. (b) The pencil of p and q.

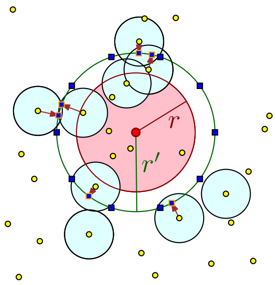


Figure 3 A slightly inaccurate depiction of how the algorithm works.

3.2 The algorithm computing the Voronoi separator

The input is a set P of n points in $\mathbb{R}^d$. The algorithms works as follows:

(A) Let $c_d = c_{\mathrm{dbl}}^d + 1$, see Definition 3. Let $\mathrm{ball}(\psi, r)$ be the smallest (closed) ball that contains n/c_d points of P where $\psi \in \mathbb{R}^d$.

(B) Pick a number r' uniformly at random from the range $[r, 2r]$.

(C) Let $b' = \mathrm{ball}(\psi, r')$.

(D) Let $P_1 = P \cap b'$ and $P_2 = P \setminus b'$.

(E) Let $\ell = r'/n^{1/d}$. Compute an ℓ-dense set Z, of size $O\Big((r'/\ell)^{d-1}\Big) = O\big(n^{1-1/d}\big)$, on the sphere $\mathbb{S} = \partial b'$ using the algorithm of Lemma 7 described below.

(F) If a point $\mathsf{p} \in P$ is in distance smaller than ℓ from $\mathbb{S}$, we insert $\mathrm{blockerSet}(\mathsf{p}, \mathbb{S})$ into the separating set Z, see Lemma 6.

We claim that the resulting set Z is the desired separator.

Efficient implementation. One can find a 2-approximation (in the radius) to the smallest ball containing n/c_d points in linear time, see [18]. This would slightly deteriorate the constants used above, but we ignore this minor technicality for the sake of simplicity of exposition. If the resulting separator is too large (i.e., larger than $\Omega\big(n^{1-1/d}\big)$ see below for details), we rerun the algorithm.

3.2.1 Computing a dense set

The following is well known, and we include it only for the sake of completeness.

▶ **Lemma 7** (proof in [7]). *Given a sphere* $\mathbb{S}$ *of radius* r' *in* $\mathbb{R}^d$*, and given a number* $\ell > 0$*, one can compute a* ℓ*-dense set* X *on* $\mathbb{S}$ *of size* $O\Big((r'/\ell)^{d-1}\Big)$*. This set can be computed in* $O(|\mathsf{X}|)$ *time.*

3.3 Correctness

▶ **Lemma 8** (proof in [7]). *We have* $|P_1| \geq n/c_d$ *and* $|P_2| \geq n/c_d$*.*

▶ **Lemma 9** (proof in [7]). *The sets* P_1 *and* P_2 *are Voronoi separated in* $\mathcal{V}(P \cup Z)$*.*

▶ **Lemma 10** (proof in [7]). *Let* $Y = |Z|$*. We have that* $\mathbf{E}[Y] \leq c_{\mathrm{sep}} n^{1-1/d}$*, where* c_{sep} *is some constant.*

3.4 The result

▶ **Theorem 11.** *Let* P *be a set of* n *points in* $\mathbb{R}^d$*. One can compute, in expected linear time, a sphere* $\mathbb{S}$*, and a set* $Z \subseteq \mathbb{S}$*, such that*

(i) $|Z| = O\big(n^{1-1/d}\big)$,
(ii) $\mathbb{S}$ *contains* $\geq cn$ *points of* P *inside it,*
(iii) *there are* $\geq cn$ *points of* P *outside* $\mathbb{S}$*, and*
(iv) Z *is a Voronoi separator of the points of* P *inside* $\mathbb{S}$ *from the points of* P *outside* $\mathbb{S}$*.*

Here $c > 0$ *is a constant that depends only on the dimension* d*.*

Proof. Clearly, each round of the algorithm takes $O(n)$ time. By Markov's inequality the resulting separator set Z is of size at most $2c_{\mathrm{sep}}n^{1-1/d}$, with probability at least $1/2$, see Lemma 10. As such, if the separator is larger than this threshold, then we rerun the algorithm. Clearly, in expectation, after a constant number of iterations the algorithm would succeed, and terminates. (It is not hard to derandomize this algorithm and get a linear running time by defining an appropriate number of concentric balls around $\mathrm{ball}(\psi, r)$ and using an averaging argument.) ◀

4 Exact algorithm for computing optimal separation of a partition

Given a set P of n points in $\mathbb{R}^d$, and a partition (P_1, P_2) of P, we are interested in computing the smallest Voronoi separating set realizing this partition.

4.1 Preliminaries and problem statement

▶ **Definition 12.** For a set $P \in \mathbb{R}^d$ and a pair of disjoint subsets (P_1, P_2), the set of *bad pairs* is $\mathcal{BP}(P, P_1, P_2) = \Big\{(\mathsf{p}_1, \mathsf{p}_2) \in P_1 \times P_2 \;\Big|\; \mathcal{C}_P(\mathsf{p}_1) \cap \mathcal{C}_P(\mathsf{p}_2) \neq \emptyset \Big\}$.

For a Voronoi diagram $\mathcal{V}(P)$, we can assume that all its faces (of various dimensions) are all triangulated (say, using bottom-vertex triangulation). This does not change the asymptotic complexity of the Voronoi diagram. For $k = 0, 1, \ldots, d$, such a k dimensional Voronoi simplex is a *k-feature*. Such a k-feature f, is induced by $d - k + 1$ sites, denoted by $\mathrm{sites}(\mathsf{f})$; that is, any point $\mathsf{p} \in \mathsf{f}$ has an equal distance to all the points of $\mathrm{sites}(\mathsf{f})$ and these are the nearest neighbor of p in P. Thus, a vertex v of the Voronoi diagram is a 0-feature, and $|\mathrm{sites}(v)| = d + 1$ (assuming general position, which we do). The *span* of a feature f, is the set of points in $\mathbb{R}^d$ that are equidistant to every site in $\mathrm{sites}(\mathsf{f})$; it is denoted by $\mathrm{span}(\mathsf{f})$ and is the k flat that contains f (it is the *affine span* of f). A *k-halfflat* is the intersection of a halfspace with a k-flat.

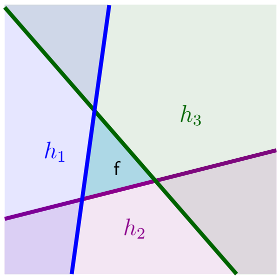


Figure 4 A 2-feature f and its induced 2-halfflats h_1, h_2, h_3.

Consider any k-feature f. The complement set $\text{span}(\mathsf{f}) \setminus \mathsf{f}$ can be covered by $k+1$ k-halfflats of span(f). Specifically, each of these halfflats is an open k-halfflat of span(f), whose boundary contains a $(k-1)$-dimensional face of the boundary of f. This set of halfflats of f, is the *shell* of f, and is denoted by shell(f), see Figure 4.

Once the Voronoi diagram is computed, it is easy to extract the "bad features". Specifically, the set of *bad features* is

$$\mathcal{F}_{\text{bad}}(P, P_1, P_2) = \left\{ \mathsf{f} \in \text{features}(\mathcal{V}(P)) \;\middle|\; \text{sites}(\mathsf{f}) \cap P_1 \neq \emptyset \text{ and } \text{sites}(\mathsf{f}) \cap P_2 \neq \emptyset \right\}.$$

Clearly, given a Voronoi diagram the set of bad features can be computed in linear time in the size of the diagram.

Given a k-feature f, it is the convex-hull of $k+1$ points; that is, $\mathsf{f} = \mathcal{CH}(\mathsf{X})$, where $\mathsf{X} = \{\mathsf{q}_1, \ldots, \mathsf{q}_{k+1}\}$. We are interested in finding the closest point in a feature to an arbitrary point p. This is a constant size problem for a fixed d, and can be solved in constant time. We denote this closest point by $\mathsf{nn}(\mathsf{p}, \mathsf{f}) = \arg\min_{\mathsf{q} \in \mathsf{f}} \mathsf{d}(\mathsf{p}, \mathsf{q})$. For the feature f, and any point p, we denote by $\text{pencil}_{\mathsf{f}}(\mathsf{p})$ the ball $\text{ball}(\mathsf{p}, \mathsf{d}(\mathsf{p}, \text{sites}(\mathsf{f})))$ (if it is uniquely defined). Furthermore, for an arbitrary set S of points, $\text{pencil}_{\mathsf{f}}(S)$ denote $\left\{ \text{ball}(\mathsf{p}, \mathsf{d}(\mathsf{p}, \text{sites}(\mathsf{f}))) \;\middle|\; \mathsf{p} \in S \right\}$. In particular, for any point $\mathsf{p} \in \mathsf{f}$, consider $\text{ball}(\mathsf{p}, \mathsf{d}(\mathsf{p}, P))$ – it contains the points of sites(f) on its boundary. The set of all such balls is the *pencil* of f, denoted by

$$\text{pencil}(\mathsf{f}) = \left\{ \text{ball}(\mathsf{p}, \mathsf{d}(\mathsf{p}, \text{sites}(\mathsf{f}))) \;\middle|\; \mathsf{p} \in \mathsf{f} \right\}. \tag{1}$$

The *trail* of f is the union of all these balls; that is, $\text{trail}(\mathsf{f}) = \bigcup_{\mathsf{p} \in \mathsf{f}} \text{ball}(\mathsf{p}, \mathsf{d}(\mathsf{p}, P))$. Finally, let mb(f) denote the smallest ball in the pencil of a feature f. Clearly, the center of mb(f) is the point $\mathsf{nn}(\mathsf{p}, \mathsf{f})$, where p is some arbitrary point of sites(f). As such, mb(f) can be computed in constant time.

▶ **Lemma 13** (proof in [7]). *Let* p *be any point and let* f *be any k-feature. The point* p *induces a halfflat of* span(f) *denoted by* $\mathcal{H}(\mathsf{p}, \mathsf{f})$*, such that* $\text{pencil}(\mathcal{H}(\mathsf{p}, \mathsf{f}))$ *is the set of all balls in* pencil(span(f)) *that contain* p*.*

We are now ready to restate our problem in a more familiar language.

▶ **Lemma 14** (Restatement of the problem {proof in [7]}). *Given a set P of n points in $\mathbb{R}^d$, and a pair of disjoint subsets (P_1, P_2), finding a minimum size Voronoi separator realizing the separation of (P_1, P_2), is equivalent to finding a minimum size hitting set of points Z, such that Z stabs (the interior) of all the balls in the set*

$$\mathcal{B} = \mathcal{B}(P, P_1, P_2) = \bigcup_{\mathsf{f} \in \mathcal{F}_{\text{bad}}(P, P_1, P_2)} \text{pencil}(\mathsf{f}). \tag{2}$$

4.2 Exact algorithm in $\mathbb{R}^d$

Given a set P of n points in $\mathbb{R}^d$, a pair of disjoint subsets (P_1, P_2) of P and an upper bound b on the number of guards, we show how one can compute a minimum size Voronoi separator realizing their separation in $n^{O(d^2\mathsf{b})}$ time. Our approach is to construct a small number of polynomial inequalities that are necessary and sufficient conditions for separation, and then use cylindrical algebraic decomposition to find a feasible solution.

▶ **Lemma 15** (proof in [7]). *For a set of guards Z, a feature f is completely removed from $\mathcal{V} = \mathcal{V}(P \cup Z)$, if and only if the induced halfflats of the guards, on $\mathrm{span}(\mathsf{f})$, cover f.*

▶ **Observation 16.** *Given a set $\mathcal{H}$ of at least $k+1$ halfflats in a k-flat, if $\mathcal{H}$ covers the k-flat, then there exists a subset of $k+1$ halfflats of $\mathcal{H}$, that covers the k-flat. This is a direct consequence of Helly's theorem[1].*

Low dimensional example. To get a better understanding of the problem at hand, the reader may imagine the subproblem of removing a bad 2-feature f (i.e. f is a triangle) from the Voronoi diagram. We know that a set Z of n guards removes f from the Voronoi diagram, if and only if the n 2-halfflats induced by Z cover the triangle f. If we add the feature-induced halfflats induced by f to the above set of halfflats, then the problem of covering the triangle f, reduces to the problem of covering the entire plane with this new set S of $n+3$ halfflats. Then from Observation 16, we have that f is removed from the Voronoi diagram if and only if there are three halfplanes of S, that cover the entire plane (there are $O(n^3)$ such triplets). Lemma 20 below shows how to convert the condition that any three 2-halfflats cover the plane, into a polynomial inequality of degree four in the coordinates of the guards.

4.2.1 Constructing the Conditions

▶ **Lemma 17** (proof in [7]). *Let f be a k-feature, and $\mathcal{H}$ be a set of k-halfflats on $\mathrm{span}(\mathsf{f})$. Then, $\mathcal{H}$ covers f $\iff$ there exists a subset $\mathcal{G} \subseteq \mathcal{H}' = \mathcal{H} \cup \mathrm{shell}(\mathsf{f})$ of size $k+1$ that covers $\mathrm{span}(\mathsf{f})$.*

▶ **Observation 18.** *Consider a set $\mathcal{H}$ of $k+1$ k-halfflats all contained in some k-flat F. We are interested in checking that $\mathcal{H}$ covers F. Fortunately, this can be done by computing the $k+1$ vertices induced by $\mathcal{H}$ on F, and verifying that each one of them is covered by the other halfflat of $\mathcal{H}$. Formally, $\mathcal{H}$ covers F $\iff$ for every halfflat $h \in \mathcal{H}$, we have that $\bigcap_{i \in \mathcal{H} \setminus \{h\}} \partial i \in h$.*

For a set P of $d+1$ points in $\mathbb{R}^d$ in general position, let $\mathrm{ball}_{\mathrm{in}}(P)$ be the unique (circumscribing) ball having all the points of P on its boundary.

▶ **Lemma 19** (proof in [7]). *Consider a k-feature f, let $\mathcal{H}$ be a set of halfflats on $\mathrm{span}(\mathsf{f})$, and let Z be a set of guards inducing the halfflats of $\mathcal{H}$. Assume that $|\mathrm{sites}(\mathsf{f})| + |Z| = d+2$. Let $h \in \mathcal{H}$ be a k-halfflat induced by a guard g, and let $\mathsf{p} = \bigcap_{i \in \mathcal{H} \setminus \{h\}} \partial i$. Then $\mathsf{p} \in h \iff \mathsf{g} \in \mathrm{ball}_{\mathrm{in}}\Big(\mathrm{sites}(\mathsf{f}) \cup \Big(Z \setminus \{\mathsf{g}\}\Big)\Big)$.*

[1] Indeed, the intersection of the complement of the halfflats of $\mathcal{H}$ is empty. As such, by Helly's theorem there exists $k+1$ of them that have an empty intersection, and the union of their complement (which are the original halfflats) cover the k-flat.

A set Z of m guards in $\mathbb{R}^d$ can be interpreted as a vector in $\mathbb{R}^{dm}$ encoding the locations of the guards. One can then reduce the requirement that Z provides the desired separation into a logical formula over the coordinates of this vector.

▶ **Lemma 20** (proof in [7]). *Let f be a bad k-feature of $\mathcal{F}_{\mathrm{bad}}(P, P_1, P_2)$, and let m be a parameter. One can compute a boolean sentence $\mathcal{A}_{\mathsf{f}}(Z)$ consisting of $m^{O(k)}$ polynomial inequalities (of degree $\leq d+2$), over dm variables, such that $\mathcal{A}_{\mathsf{f}}(Z)$ is true $\iff$ the set of m guards Z (induced by the solution of this formula) destroys f completely when inserted. Formally, for every $\mathsf{f}' \in \mathcal{F}_{\mathrm{bad}}(P \cup Z, P_1, P_2)$, we have $\mathsf{f} \cap \mathsf{f}' = \emptyset$.*

4.2.2 The Result

▶ **Theorem 21.** *Let P be a set of n points in $\mathbb{R}^d$. Let (P_1, P_2) be some disjoint partition of P and b be a parameter, such that there exists a Voronoi separator for (P_1, P_2) of at most b points. A minimum size Voronoi separator can be computed in $n^{O(d^2\mathsf{b})}$ time, if such a separator exists of size at most b.*

Proof. Let g be a parameter to be fixed shortly. Let Z be a set of g guards in $\mathbb{R}^d$. By Lemma 20, the condition $\mathcal{A}(Z)$ that every bad feature is removed from the Voronoi diagram of $P \cup Z$, can be written as

$$\mathcal{A}(Z) = \bigwedge_{\mathsf{f} \in \mathcal{F}_{\mathrm{bad}}(P, P_1, P_2)} \mathcal{A}_{\mathsf{f}}(Z).$$

Namely, Z is a Voronoi separator for (P_1, P_2). The formula $\mathcal{A}(Z)$ contains $n^{O(d)}$ degree-$(d+2)$ polynomial inequalities comprising of at most $d\mathrm{g}$ variables. By [5, Theorem 13.12], one can compute, in $n^{O(d^2\mathrm{g})}$ time, a solution as well as the sign of each polynomial in $\mathcal{P}$, for every possible sign decomposition that $\mathcal{P}$ can attain. Now for each attainable sign decomposition, we simply check if $\mathcal{A}(Z)$ is true. This can be done in $n^{O(d)}$ time. The algorithm computes a Voronoi separator for (P_1, P_2) in P of size g, in $n^{O(d^2\mathrm{g})}$ time, if such a separator exists.

Now, the algorithms tries $\mathrm{g} = 1, \ldots, \mathsf{b}$ and stops as soon as a feasible solution is found. ◀

5 Constant factor approximation

Given a set P of n points in $\mathbb{R}^d$, and a partition (P_1, P_2) of P, we show how one can compute in $n^{O(d)}$ time, a Voronoi separator Z for (P_1, P_2), whose size is a constant factor approximation to the size of an optimal Voronoi separator realizing such a partition.

▶ **Theorem 22** (proof in [7]). *Let P be a set of n points in $\mathbb{R}^d$, and (P_1, P_2) be a given pair of disjoint subsets of P. One can compute a Voronoi separator Z that realizes this partition. The algorithm runs in $O(n^2 \log n)$ time for $d = 2$, and in $O\big(n^{\lceil d/2 \rceil + 1}\big)$ time, for $d > 2$. The algorithm provides a constant factor approximation to the smallest Voronoi separator realizing this partition.*

6 Stronger Separator for PTAS

We are interested in a $(1+\varepsilon)-$approximation to an optimal Voronoi separator of a given partition. As implied by Lemma 14, this boils down to a hitting set problem over balls. The challenge here is that the set of balls is an infinite set, see Eq. (2). We need the following improved separator theorem, whose proof can be found in the full version [7].

▶ **Theorem 23** (proof in [7]). *Let* X *be a set of points in* $\mathbb{R}^d$*, and* $k > 0$ *be an integer. One can compute, in* $O(|\mathsf{X}|)$ *expected time, a set* Z *of* $O\bigl(k^{1-1/d}\bigr)$ *points and a sphere* $\mathbb{S}$ *containing* $\Theta(k)$ *points of* X *inside it, such that for any set* $\mathcal{B}$ *of balls stabbed by* X*, we have that every ball of* $\mathcal{B}$ *that intersects* $\mathbb{S}$ *is stabbed by a point of* Z*.*

7 Local Search PTAS for geometric hitting set

Here, we present a simple local search $(1+\varepsilon)$-approximation algorithm (PTAS) for hitting set problems of balls (or fat objects) in $\mathbb{R}^d$. The set of balls is either specified explicitly, or as in the case of the Voronoi partition, implicitly.

7.1 The algorithm LocalHitBalls

7.1.1 Preliminaries

Let $\mathcal{B}$ be a set of balls (possibly infinite) in $\mathbb{R}^d$ that is represented (potentially implicitly) by an input of size n, and let $\nu(\mathcal{B})$ be the size of its minimum hitting set. We assume that $\mathcal{B}$ satisfies the following properties:

(A) **Initial solution**: One can compute a hitting set X_0 of $\mathcal{B}$ in $n^{O(1)}$ time, such that the size of X_0 is a constant factor approximation to the optimal.

(B) **Local exchange**: Let X and Y be point sets, such that
  - **(i)** X is a hitting set of $\mathcal{B}$ (i.e., it is the current solution),
  - **(ii)** $|\mathsf{X}| \leq n^{O(1)}$ (i.e., it is not too large),
  - **(iii)** $\mathsf{Y} \subseteq \mathsf{X}$ (i.e., Y is the subset to be replaced),
  - **(iv)** $|\mathsf{Y}| \leq \ell$, where ℓ is any integer.

  Then one can compute in $n^{O(\ell)}$ time, the smallest set Y', such that $(\mathsf{X} \setminus \mathsf{Y}) \cup \mathsf{Y}'$ is a hitting set of $\mathcal{B}$.

7.1.2 Algorithm

The input is a set $\mathcal{B}$ of balls in $\mathbb{R}^d$ satisfying the properties above. The algorithm works as follows:

(A) Compute an initial hitting set X_0 of $\mathcal{B}$ (see (P1)), and set $\mathsf{X} \leftarrow \mathsf{X}_0$.

(B) While there is a beneficial exchange of size $\leq \ell = O\bigl(1/\varepsilon^d\bigr)$ in X, carry it out using (P2). Specifically, verify for every subset $\mathsf{Y} \subseteq \mathsf{X}$ of size at most ℓ, if it can be replaced by a strictly smaller set Y' such that $(\mathsf{X} \setminus \mathsf{Y}) \cup \mathsf{Y}'$ remains a hitting set of $\mathcal{B}$. If so, set $\mathsf{X} \leftarrow (\mathsf{X} \setminus \mathsf{Y}) \cup \mathsf{Y}'$, and repeat.

(C) If no such set exists, then return X as the hitting set.

Details. For the geometric hitting set problem where $\mathcal{B}$ is a set of n balls in $\mathbb{R}^d$, (P1) follows by a simple greedy algorithm hitting the smallest ball (in the spirit of Theorem 22) – see also [9]. As for (P2), one can check for a smaller hitting set of size at most ℓ by computing the arrangement $\mathcal{A}(\mathcal{B})$, and directly enumerating all possible hitting sets of size at most ℓ.

The more interesting case is when the set $\mathcal{B}$ is defined implicitly by an instance of the minimum Voronoi separation problem (i.e., we have a set P of n points in $\mathbb{R}^d$, and a desired Voronoi partition (P_1, P_2) of P). Then (P1) follows from Theorem 22. Furthermore, (P2) follows from the algorithm of Theorem 21 through computing the minimum hitting set Y' of $\mathcal{B}(P \cup (\mathsf{X} \setminus \mathsf{Y}), P_1, P_2)$, in $n^{O(\ell)}$ time, where $|\mathsf{Y}'| \leq \ell$.

```
ClusterLocalOpt(𝓑, 𝓛, 𝓞):
    // 𝓑:  given set of balls need hitting.
    // 𝓛 ⊆ ℝ^d:  local optimum set hitting 𝓑 computed by LocalHitBalls.
    // 𝓞 ⊆ ℝ^d:  optimal hitting set.
  k := O(1/ε^d),   L_1 := 𝓛,   O_1 := 𝓞,   Z_1 := ∅,   𝓑_1 := 𝓑,   and i := 1.
  while L_i ∪ Z_i ≠ ∅ do
      Apply Theorem 23 to L_i ∪ Z_i with the parameter min(k, |L_i ∪ Z_i|).
      b_i, T_i: ball and the separator set returned, respectively.
      f_i := b_i \ ⋃_{j=1}^{i-1} b_j be the region of space newly covered by b_i.
      𝓛_i := 𝓛 ∩ f_i and 𝓞_i = 𝓞 ∩ f_i.
      Z_{i+1} := (Z_i \ b_i) ∪ T_i.
      L_{i+1} := L_i \ 𝓛_i.
      i := i + 1
  I := i.
```

Figure 5 Clustering the local optimal solution.

7.2 Quality of approximation

The bound on the quality of approximation follows by a clustering argument similar to our bottom-up PTAS algorithm (see full version [7]). The clustering is described in Figure 5. This simultaneously breaks up the local optimal solution $\mathcal{L}$ and the optimal solution $\mathcal{O}$ into small clusters, and we will use a local exchange argument to bound their corresponding sizes. The following lemma testifies that this clustering algorithm provides a "smooth" way to move from $\mathcal{L}$ to $\mathcal{O}$, through relatively small steps.

▶ **Lemma 24.** *Any ball of $\mathcal{B}$ is stabbed by $\mathsf{X}_{i+1} = L_{i+1} \cup Z_{i+1} \cup \bigcup_{j=1}^{i} \mathcal{O}_j$, for all i, where $L_{i+1} = \bigcup_{j=i+1}^{I} \mathcal{L}_i$.*

Proof. The claim holds clearly for $i = 0$, as $L_1 = \mathcal{L}$, where $\mathcal{L}$ is the locally optimal solution. Now, for the sake of contradiction, consider the smallest i for which the claim fails, and let $b \in \mathcal{B}$ be the ball that is not being stabbed. We have that b is stabbed by $\mathsf{X}_i = \left(\bigcup_{j=i}^{I} \mathcal{L}_i\right) \cup Z_i \cup \left(\bigcup_{j=1}^{i-1} \mathcal{O}_j\right)$ but not by X_{i+1}.

It can not be that b is stabbed by a point of $\bigcup_{j=1}^{i-1} \mathcal{O}_j$, as such a point is also present in X_{i+1}. As such, b must be stabbed by one of the points of $L_i \cup Z_i$, and then this stabbing point must be inside b_i – indeed, the points removed from L_i and Z_i as we compute L_{i+1} and Z_{i+1}, respectively, are the ones inside b_i. Now, if b intersects ∂b_i, then $T_i \subseteq Z_{i+1}$ stabs b by Theorem 23, a contradiction.

So it must be that $b \subseteq b_i$. But then consider the region $F_i = \cup_{j=1}^{i} f_j = \cup_{j=1}^{i} b_j$. It must be that $b \subseteq F_i$. This in turn implies that the point of $\mathcal{O}$ stabbing b is in $\mathcal{O} \cap F_i = \cup_{j=1}^{i} \mathcal{O}_i \subseteq \mathsf{X}_{i+1}$. A contradiction. ◀

▶ **Lemma 25.** *Consider any ball $b \in \mathcal{B}$. Let i be the smallest index such that $\mathcal{O}_i$ stabs b. Then b is stabbed by $\mathcal{L}_i \cup T_i \cup (Z_i \cap b_i)$. Furthermore for all i, $o_i \leq \lambda_i + t_i + z_i$ and $o_i = O(k)$, where $o_i = |\mathcal{O}_i|$, $t_i = |T_i|$, $\lambda_i = |\mathcal{L}_i|$, and $z_i = |Z_i \cap b_i|$.*

Proof. If b intersects ∂b_i, then it is stabbed by T_i by Theorem 23. Otherwise, $b \subseteq F_i = \cup_{j=1}^{i} f_j = \cup_{j=1}^{i} b_j$. In particular, this implies that no later point of $\mathcal{O}_{i+1} \cup \ldots \cup \mathcal{O}_t$ can stab b. That is, only $\mathcal{O}_i$ stabs b. By Lemma 24 both X_i and X_{i+1} stab b, where $\mathsf{X}_i =$

$(\mathcal{L} \setminus F_{i-1}) \cup Z_i \cup \bigcup_{j=1}^{i-1} \mathcal{O}_j$. Namely, b is stabbed by a point of $(\mathcal{L} \setminus F_{i-1}) \cup Z_i$ that is contained inside b_i. Such a point is either in $Z_i \cap b_i$, or in $(\mathcal{L} \setminus F_{i-1}) \cap b_i = \mathcal{L}_i$, as claimed.

The second part follows by observing that otherwise $\mathcal{O}_i$ can be replaced by $\mathcal{L}_i \cup T_i \cup (Z_i \cap b_i)$ in the optimal solution.

The third claim follows by observing that by the algorithm design $\lambda_i + z_i = O(k)$, and $t_i = O\big(k^{1-1/d}\big)$. As such, $o_i \leq \lambda_i + t_i + z_i = O(k)$. ◀

▶ **Lemma 26.** *Consider any ball $b \in \mathcal{B}$ that is stabbed by $\mathcal{L}_i$ but it is not stabbed by $\mathcal{L} \setminus \mathcal{L}_i$. Then the ball b is stabbed by $\mathcal{O}_i \cup T_i \cup (Z_i \cap b_i)$. Additionally, using the notations of Lemma 25, we have $\lambda_i \leq o_i + t_i + z_i$.*

Proof. If b intersects ∂b_i, then it is stabbed by T_i by Theorem 23. So we assume for now on that b is not stabbed by T_i. But then, $b \subseteq b_i \subseteq F_i = \cup_{j=1}^{i} f_j = \cup_{j=1}^{i} b_j$.

Now, by Lemma 24 both X_i and X_{i+1} stab b, where $\mathsf{X}_{i+1} = \bigcup_{j=i+1}^{I} \mathcal{L}_j \cup Z_{i+1} \cup \bigcup_{j=1}^{i} \mathcal{O}_j$. Specifically, b is stabbed by a point of $Z_{i+1} \cup \bigcup_{j=1}^{i} \mathcal{O}_j$ that is contained inside b_i. Such a point is either in $Z_i \cap b_i$ and then we are done, or alternatively, it can be in $b_i \cap \bigcup_{j=1}^{i} \mathcal{O}_j$.

Now, if $b \subseteq f_i$ then $\mathcal{O}_i$ must stab b, and we are done. Otherwise, let $k < i$ be the maximum index such that b_i intersects ∂b_k. Observe that as b intersects f_i, it can not be that it intersects the balls $b_{k+1}, \ldots, b_{i-1}$. In particular, Theorem 23 implies that there is a point of T_k that stabs b, as b is being stabbed by $L_k \cup Z_k$. This point of T_k is added to Z_{k+1}, and it is not being removed till Z_{i+1}. As such, this point is in Z_i, and it is also in b_i, thus implying the claim.

As for the second part, by Lemma 25, $o_i + t_i + z_i \leq \alpha = O(k)$. As such, setting $\ell = \Omega(1/\varepsilon^d)$ to be sufficiently large, we have that $\ell > 2\alpha$, and the local search algorithm would consider the local exchange of $\mathcal{L}_i$ with $\mathcal{O}_i \cup T_i \cup (Z_i \cap b_i)$. As this is an exchange not taken, it must be that it is not beneficial, implying the inequality. ◀

▶ **Lemma 27.** *We have that* (i) $\sum_{i=1}^{I} t_i \leq (\varepsilon/4)\,|\mathcal{O}|$, *and* (ii) $\sum_{i=1}^{I} z_i \leq (\varepsilon/4)\,|\mathcal{O}|$.

Proof. Observe that $k = O(1/\varepsilon^d)$ and $t_i = |T_i| = O(k^{1-1/d}) = O\big(1/\varepsilon^{d-1}\big) \leq c\varepsilon k$, where c can be made arbitrarily small by making k sufficient large. In particular, in every iteration, the algorithm removes $\geq k$ points from $L_i \cup Z_i$, and replaces them by t_i points. Starting with a solution of size $|\mathcal{L}| \leq |\mathsf{X}_0| \leq c'\,|\mathcal{O}|$, where c' is some constant, this can happen at most $I \leq c'\,|\mathcal{O}|\,/(k - c\varepsilon k) = O(\varepsilon^d\,|\mathcal{O}|)$. As such, we have $\sum_{i=1}^{I} t_i = O\big(\varepsilon^d\,|\mathcal{O}|\,c\varepsilon k\big) = O(c\varepsilon\,|\mathcal{O}|)$, as $k = O(1/\varepsilon^d)$. The claim now follows by setting k to be sufficiently large.

The second claim follows by observing that t_i counts the numbers of points added to Z_{i+1}, while z_i counts the number of points removed from it. As Z_I is empty, it must be that $\sum_i t_i = \sum_i z_i$. ◀

▶ **Lemma 28.** *We have that $|\mathcal{L}| \leq (1+\varepsilon)\,|\mathcal{O}|$.*

Proof. We have by Lemma 26 and Lemma 27 that $|\mathcal{L}| = \sum_{i=1}^{I} \lambda_i \leq \sum_{i=1}^{I}(o_i + t_i + z_i) = |\mathcal{O}| + \sum_{i=1}^{I} t_i + \sum_{i=1}^{I} z_i \leq (1+\varepsilon/2)\,|\mathcal{O}|$. ◀

7.2.1 The Result

▶ **Theorem 29.** *Let $\mathcal{B}$ be a set of balls in $\mathbb{R}^d$ satisfying properties (P1) and (P2). Then* ***LocalHitBalls*** *computes, in $n^{O(1/\varepsilon^d)}$ time, a hitting set of $\mathcal{B}$, whose size is a $(1+\varepsilon)$-approximation to a minimum size hitting set of $\mathcal{B}$.*

Proof. Lemma 28 implies the bound on the quality of approximation.

As for the runtime, observe that the size of local solution reduces by at least one after each local improvement, and from (P1), the initial local solution has size $n^{O(1)}$. Thus there can be at most $n^{O(1)}$ local improvement steps before the algorithm stops. Furthermore, every local solution has at most $n^{O(\ell)}$ subsets of size ℓ that are checked for local improvement. By (P2), such a local improvement can be checked in $n^{O(\ell)}$ time. Thus **LocalHitBalls** runs in $n^{O\left(1/\varepsilon^d\right)}$ time. ◀

8 Conclusions

We presented a new separator result that provides a new way to perform geometric divide and conquer for Voronoi diagrams (or Delaunay triangulations). We use this to derive a new PTAS for the Voronoi partition problem, making progress on a geometric hitting set problem where the ranges to be hit are defined implicitly, and their number is infinite. Significantly, the resulting local search algorithm is relatively simple, and should have other applications [4].

There are many interesting open problems for further research. In particular, the new PTAS might be more practical for the piercing balls problem than previous algorithms, and it might be worthwhile to further investigate this direction. Additionally, the proof technique for the local search algorithm might be applicable to other separator based geometric problems.

References

1 N. Alon, P. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. In *J. Amer. Math. Soc.*, volume 3, pages 801–808, 1990. doi:10.1090/S0894-0347-1990-1065053-0.

2 V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004. doi:10.1137/S0097539702416402.

3 F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. doi:10.1142/8685.

4 S. Bandyapadhyay and K. R. Varadarajan. On variants of k-means clustering. *CoRR*, abs/1512.02985, 2015. Also in SoCG 16. URL: http://arxiv.org/abs/1512.02985.

5 S. Basu, R. Pollack, and M. F. Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer, 2006. doi:10.1007/3-540-33099-2.

6 M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Santa Clara, CA, USA, 3rd edition, 2008. URL: http://www.cs.uu.nl/geobook/.

7 V. V. S. P. Bhattiprolu and S. Har-Peled. Separating a voronoi diagram via local search. *CoRR*, abs/1401.0174, 2014. URL: http://arxiv.org/abs/1401.0174.

8 J. Böttcher, K. P. Pruessmann, A. Taraz, and A. Würfl. Bandwidth, expansion, treewidth, separators and universality for bounded-degree graphs. *Eur. J. Comb.*, 31(5):1217–1227, July 2010. doi:10.1016/j.ejc.2009.10.010.

9 T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003. doi:10.1016/S0196-6774(02)00294-8.

10 T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete Comput. Geom.*, 48:373–392, 2012. doi:10.1007/s00454-012-9417-5.

11 Z. Dvorak and S. Norin. Strongly sublinear separators and polynomial expansion. *ArXiv e-prints*, April 2015. URL: http://arxiv.org/abs/1504.04821, arXiv:1504.04821.

12 A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Inform. Process. Lett.*, 100(6):238–245, 2006. doi:10.1016/j.ipl.2006.05.014.

13 D. Eisenstat and P. N. Klein. Linear-time algorithms for max flow and multiple-source shortest paths in unit-weight planar graphs. In *Proc. 45th Annu. ACM Sympos. Theory Comput.* (STOC), pages 735–744, 2013. doi:10.1145/2488608.2488702.

14 T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005. doi:10.1137/S0097539702402676.

15 J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *J. Comput. Sys. Sci.*, 72(5):868–889, 2006. doi:10.1016/j.jcss.2005.05.007.

16 J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan. A separator theorem for graphs of bounded genus. *J. Algorithms*, 5(3):391–407, 1984. doi:10.1016/0196-6774(84)90019-1.

17 A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *ArXiv e-prints*, September 2008. URL: http://arxiv.org/abs/0809.2554, arXiv:0809.2554.

18 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., Boston, MA, USA, 2011. URL: http://sarielhp.org/book/, doi:10.1090/surv/173.

19 S. Har-Peled. A simple proof of the existence of a planar separator. *ArXiv e-prints*, 2011. URL: http://arxiv.org/abs/1105.0103, arXiv:1105.0103.

20 S. Har-Peled and M. Lee. Weighted geometric set cover problems revisited. *J. Comput. Geom.*, 3(1):65–85, 2012. URL: http://sarielhp.org/papers/08/expand_cover/.

21 S. Har-Peled and K. Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. In *Proc. 23nd Annu. Euro. Sympos. Alg.* (ESA), volume 9294 of *Lect. Notes in Comp. Sci.*, pages 717–728, 2015. doi:10.1007/978-3-662-48350-3_60.

22 P. N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562.

23 P. Koebe. Kontaktprobleme der konformen Abbildung. *Ber. Verh. Sächs. Akademie der Wissenschaften Leipzig, Math.-Phys. Klasse*, 88:141–164, 1936.

24 R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36:177–189, 1979. doi:10.1137/0136016.

25 R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980. doi:10.1137/0209046.

26 G. L. Miller, S. H. Teng, W. P. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. Assoc. Comput. Mach.*, 44(1):1–29, 1997. doi:10.1145/256292.256294.

27 N. H. Mustafa, R. Raman, and S. Ray. Settling the APX-hardness status for geometric set cover. In *Proc. 55th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 541–550, 2014. doi:10.1109/FOCS.2014.64.

28 N. H. Mustafa and S. Ray. Improved results on geometric hitting set problems. *Discrete Comput. Geom.*, 44(4):883–895, 2010. doi:10.1007/s00454-010-9285-9.

29 J. Nešetřil and P. Ossona de Mendez. Grad and classes with bounded expansion I. decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008.

30 J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, 1987. URL: http://cs.smith.edu/~orourke/books/art.html.

31 J. Pach and P. K. Agarwal. *Combinatorial Geometry*. John Wiley & Sons, 1995. URL: http://www.addall.com/Browse/Detail/0471588903.html.

32 W. D. Smith and N. C. Wormald. Geometric separator theorems and applications. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 232–243, 1998. doi:10.1109/SFCS.1998.743449.

33 C. Sommer, E. Verbin, and W. Yu. Distance oracles for sparse graphs. In *Proc. 50th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 703–712, Georgia, USA, 2009. doi:10.1109/FOCS.2009.27.

34 P. Ungar. A theorem on planar graphs. *J. London Math. Soc.*, 26:256–262, 1951.