Submodular Clustering in Low Dimensions

Arturs Backurs
Toyota Technological Institute at Chicago, Il, USA

Sariel Har-Peled
University of Illinois at Urbana-Champaign, Il, USA

Abstract

We study a clustering problem where the goal is to maximize the coverage of the input points by k chosen centers. Specifically, given a set of n points $P \subseteq \mathbb{R}^d$, the goal is to pick k centers $C \subseteq \mathbb{R}^d$ that maximize the service $\sum_{p \in P} \varphi\big(\mathsf{d}(p, C)\big)$ to the points P, where $\mathsf{d}(p, C)$ is the distance of p to its nearest center in C, and φ is a non-increasing service function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$. This includes problems of placing k base stations as to maximize the total bandwidth to the clients – indeed, the closer the client is to its nearest base station, the more data it can send/receive, and the target is to place k base stations so that the total bandwidth is maximized. We provide an $n^{\varepsilon^{-O(d)}}$ time algorithm for this problem that achieves a $(1-\varepsilon)$-approximation. Notably, the runtime does not depend on the parameter k and it works for an arbitrary non-increasing service function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$.

2012 ACM Subject Classification Theory of computation → Facility location and clustering; Theory of computation → Computational geometry

Keywords and phrases clustering, covering, PTAS

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.8

Funding *Sariel Har-Peled*: Work on this paper was partially supported by a NSF AF awards CCF-1907400 and CCF-1421231.

1 Introduction

Clustering is a fundamental problem, used almost everywhere in computing. It involves partitioning the data into groups of similar objects – and, under various disguises, it is the fundamental problem underlying most machine learning applications. The (theoretically) well studied variants include k-median, k-means and k-center clustering. But many other variants of the clustering problem have been subject of a long line of research [10].

A clustering problem is often formalized as a constrained minimization problem of a cost function. The cost function captures the similarity of the objects in the same cluster. By minimizing the cost function we obtain a clustering of the data such that objects in the same cluster are more similar (in some sense) to each other than to those in other clusters. Many of this type of formalizations of clustering are both computationally hard, and sensitive to noise – often because the number of clusters is a hard constraint.

Clustering as a quality of service maximization

An alternative formalization of the clustering problem is as a *maximization* problem where the goal is to maximize the quality of "service" the data gets from the facilities chosen. As a concrete example, consider a set of n clients, and the problem is building k facilities. The quality of service a client gets is some monotonically decreasing non-negative function of its distance to the closest facility. As a concrete example, for a mobile client, this quantity might be the bandwidth available to the client. We refer to this problem as the *k-service* problem.

Such a formalization of clustering has several advantages. The first is diminishing returns (a.k.a. submodularity) – that is, the marginal value of a facility decreases as one adds more facilities. This readily leads to an easy constant approximation algorithm. A second

significant advantage is the insensitivity to outliers – a few points being far away from the chosen facilities are going to change the target function by an insignificant amount (naturally, these outliers would get little to no service).

Formal problem statement: k-service

Given a set $P \subseteq \mathbb{R}^d$ of n points, a monotonically decreasing function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$, the goal is to choose a set C of k centers (not necessarily among the n given points), that maximize $\sum_{p \in P} \varphi\big(\mathsf{d}(p, C)\big)$, where $\mathsf{d}(p, C) = \min_{c \in C} \|p - c\|$.

Our result

We obtain an $n^{\varepsilon^{-O(d)}}$ time algorithm for this problem that achieves $(1 - \varepsilon)$-approximation for points in $\mathbb{R}^d$.

Related work

Maximum coverage problems, such as partial coverage by disks, were studied in the past [15]. These problems can be interpreted as a k-service problem, where the function is 1 within distance r from a facility, and zero otherwise. In particular, Chaplick *et al.* [3] showed a PTAS for covering a maximum number of points, out of a given set of disks in the plane. Our result implies a similar result in higher dimensions, except that we consider the continuous case (i.e., our results yields a PTAS for covering the maximum number of points using k unit disks). Cohen-Addad *et al.* [7] showed that local search leads to a PTAS for k-median and k-means clustering in low dimensions (and also in minor-free graphs). In [5] it was shown that the local search for k-means can be made faster achieving the runtime of $n \cdot k \cdot (\log n)^{(d/\varepsilon)^{O(d)}}$. In [6] near-linear time approximation schemes were obtained for several clustering problems improving on an earlier work (in particular, [12]). The authors achieve the runtime of $2^{(1/\varepsilon)^{O(d^2)}} n(\log n)^{O(1)}$. The techniques from the above works do not seem to be able to give near-linear time solution for the k-service problem, unfortunately. For instance, consider the special case of the k-service problem with $k = 1$ and where the service function is 1 within distance r from a facility, and zero otherwise (the maximum coverage problem as above). Even for this very special case of the problem there is no algorithm known running in time $n^{o(d)}$ where d is the dimension of the underlying space. The special case of $k = 1$ is a significant obstacle towards obtaining near-linear time algorithms for the k-service problem.

Another related line of work is on the kernel density estimation problem where a set P of n points is given and the goal is to preprocess P such that for an arbitrary query point c one can efficiently approximate $\sum_{p \in P} \varphi\big(\mathsf{d}(p, c)\big)$. The goal is to answer such queries much faster than in $O(nd)$ time, which is just the linear scan. For various service functions φ and distance functions d significantly faster algorithms are known [14, 4, 1]. Despite the similarity, however, finding a point (center) c that (approximately) maximizes the sum $\sum_{p \in P} \varphi\big(\mathsf{d}(p, c)\big)$ seems to be a much harder problem [13].[1] Our work can be seen as a generalization of the latter problem where our goal is to pick k centers instead of one center.

In a another work, Friggstad *et al.* [11] addressed the clustering problem in the setting with outliers.

[1] In particular, to find such a point, [13] use a heuristic that iteratively computes the gradient (mean shift vector) to obtain a sequence of points that converge to a local maxima (mode) of the density.

Balanced divisions

One of the building blocks we need is balanced divisions for Voronoi diagrams. This is present in the recent paper of Cohen-Addad *et al.* [7]. The idea of balanced divisions seems to go back to the work of Cohen-Addad and Mathieu [8]. Chaplick *et al.* [3] also prove a similar statement for planar graphs.

For the sake of completeness, we include the proof of the desired balanced divisions we need in Appendix A. Both here and [7] uses the Voronoi separator of Bhattiprolu and Har-Peled [2] as the starting point to construct the desired divisions. The Voronoi divisions we construct here are slightly stronger than the one present in [7] – all the batches in the division are approximately of the same size, and each one has a small separator from the rest of the point set.

Clustering and submodularity

Work using submodularity in clustering includes Nagano *et al.* [16] and Wei *et al.* [17]. Nagano *et al.* [16] considers the problem of computing the multi-way cut, that minimizes the average cost (i.e., number of edges in the cut divided by the number of clusters in the cut). Wei *et al.* [17] also studies such partitions with average cost target function. These works do not have any direct connection to what is presented here, beyond the usage of submodularity.

Paper organization

We define the problem formally in Section 2, and review some necessary tools including submodularity and balanced subdivisions. Section 3 describes how to find a good exchange for the current solution. We describe the local search algorithm in Section 4. The main challenge is to prove that if the local search did not reach a good approximation, then there must be a good exchange that improves the solution – this is proved in Section 5.

2 Preliminaries

Notations

In the following, we use $X+x$ and $X-x$ as a shorthand for $X\cup\{x\}$ and $X\setminus\{x\}$, respectively.

Given a point $p\in\mathbb{R}^d$, and a set $D\subseteq\mathbb{R}^d$, we denote by $\mathsf{d}(p,D)=\min_{f\in D}\|p-f\|$ the *distance* of p from D. A point in D realizing this distance is the *nearest-neighbor* to p in D, and is denoted by $\mathsf{nn}(p,D)=\arg\min_{c\in D}\|c-p\|$.

2.1 Service function and problem statement

A service function is a monotonically non-increasing function $\varphi:\mathbb{R}^+\to\mathbb{R}^+$. In the following, given $x\geq 0$, assume that one can compute, in constant time, both $\varphi(x)$ and $\varphi^{-1}(x)$. Given a point $p\in\mathbb{R}^d$, and a center $c\in\mathbb{R}^d$, the quality of service that c provides p is $\rho(c,p)=\varphi\big(\|p-c\|\big)$. For a set of centers C, the quality of service it provides to p is

$$\rho(C,p)=\max_{c\in C}\rho(c,p)=\rho(\mathsf{nn}(p,C),p).$$

The service to P provided by the set of centers C, or just profit, is

$$\rho(C)=\rho(C,P)=\sum_{p\in P}\rho(C,p).$$

In the k-service problem, the task is to compute the set C^* of k points that realizes

$$\mathrm{opt}_k(P)=\max_{C\subseteq\mathbb{R}^d,|C|=k}\rho(C,P).$$

2.2 Submodularity

The above is a submodular optimization problem. Indeed, consider a center point c, and a set of centers C. The cell of c, in the Voronoi partition induced by C, is

$$\mathsf{cl}(c, C) = \left\{ p \in P \;\middle|\; \|c - p\| < \mathsf{d}(p, C - c) \right\}.$$

▶ **Definition 1.** *The marginal value of c is*

$$\nabla(c, C) = \rho(C + c, P) - \rho(C - c, P) = \sum_{p \in \mathsf{cl}(c, C+c)} (\rho(C + c, p) - \rho(C - c, p)).$$

In words, this is the increase in the service that one gets from adding the center c.

For two sets of centers $D \subseteq C$, and a center c, observe that $\mathsf{cl}(c, C + c) \subseteq \mathsf{cl}(c, D + c)$. In particular, we have

$$\begin{aligned}\nabla(c, D) &= \sum_{p \in \mathsf{cl}(c, D+c)} (\rho(c, p) - \rho(D, p)) \geq \sum_{p \in \mathsf{cl}(c, C+c)} (\rho(c, p) - \rho(D, p)) \\ &\geq \sum_{p \in \mathsf{cl}(c, C+c)} (\rho(c, p) - \rho(C, p)) = \nabla(c, C).\end{aligned}$$

This property is known as submodularity.

2.3 Balanced divisions

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 $p \in P$ is

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

where $\mathsf{d}(q, P) = \min_{t \in P} \|q - t\|$ is the distance of q to the set P, see [9] for more details on Voronoi diagrams.

▶ **Definition 2.** *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 $p_1 \in P_1$ and $p_2 \in P_2$, we have that their Voronoi cells are disjoint – that is, $\mathcal{C}_P(p_1) \cap \mathcal{C}_P(p_2) = \emptyset$. That is, the Voronoi cells of the pointsets are non-adjacent.*

▶ **Definition 3.** *Given a set P of n points in $\mathbb{R}^d$, a set of pairs $\{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$ is a Voronoi α-division of P, if for all i, we have*

- **(i)** *$B_1, \ldots, B_m$ are disjoint,*
- **(ii)** *$\bigcup_j B_j = P$,*
- **(iii)** *pointset ∂_i Voronoi separates B_i from $P \setminus B_i$ in the Voronoi diagram of $P \cup \partial_i$ in the sense of Definition 2, and*
- **(iv)** *$|B_i| \leq \alpha$.*

The set B_i is the ith batch, and ∂_i is its boundary.

A balanced coloring is a coloring $\chi : P \to \{-1, +1\}$ of P by ± 1, such that $\chi(P) = \sum_{p \in P} \chi(p) = 0$. For a set $X \subseteq P$, its discrepancy is $|\chi(X)|$. We need the following balanced α-division result. Since this result is slightly stronger than what is available in the literature, and is not stated explicitly in the form we need it, we provide a proof for the sake of completeness in Appendix A.

▶ **Theorem 4.** *Given a set P of n points in $\mathbb{R}^d$, parameters $\delta \in (0,1)$ and $\alpha = \Omega(1/\delta^{d+1})$, and a balanced coloring χ of P, one can compute in polynomial time, a Voronoi α-division $\mathcal{D} = \{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$, such that the following holds:*

(A) $\bigcup B_i = P$, *and the batches $B_1, \ldots, B_m$ are disjoint.*

(B) $m = O(n/\alpha)$.

(C) *For all i, we have the following properties:*

- **(C.i)** *the set ∂_i Voronoi separates B_i from $P \setminus B_i$.*
- **(C.ii)** $(1-\delta)\alpha \leq |B_i| \leq \alpha$ *(except for the last batch, which might be of size at least $(1-\delta)\alpha$, and at most size 2α).*
- **(C.iii)** $|\partial_i| \leq \delta\,|B_i|$.
- **(C.iv)** $|\chi(B_i)| \leq \delta\,|B_i|$.

3 Computing a good local exchange (if it exists)

▶ **Lemma 5** (Computing a good single center)**.** *Let P be a set of n points in $\mathbb{R}^d$, and let C be a set of k centers. Given a parameter $\varepsilon \in (0,1)$, one can $(1-\varepsilon)$-approximate the center $c \in \mathbb{R}^d$ that maximizes the marginal value in $(n/\varepsilon)^{O(d)}$ time. Formally, we have $\nabla(c, C) \geq (1-\varepsilon)\max_{f \in \mathbb{R}^d \setminus C} \nabla(f, C)$.*

Proof. Let $\rho = \rho(C, P)$, $g = \arg\max_{f \in P} \nabla(f, C)$, $\Delta = \nabla(g, C)$, and $\mathsf{u} = \varphi(0)$. Clearly, the profit of the optimal solution, after adding any number of centers to C (but at least one), is somewhere in the interval $[\rho + \Delta, n\mathsf{u}] \subseteq [\rho + \Delta, \rho + n\Delta]$, which follows from $\mathsf{u} \leq (\rho/n) + \Delta$. For a point $p \in P$, let

$$v(p,i) = \min\Big(\rho(C,p) + \ell(i),\, \mathsf{u}\Big) \qquad \text{where } \ell(i) = (1+\varepsilon/4)^i \frac{\varepsilon\Delta}{4n},$$

for $i = 0, \ldots, N$, where $N = \lceil 16(\ln n)/\varepsilon^2 \rceil$. Let $r(p,i) = \varphi^{-1}\big(v(p,i)\big)$ (this is the radius from p where a center provides service $v(p,i)$).

Place a sphere of radius $r(p,i)$ around each point $p \in P$, for $i = 0, \ldots, N$. Let $\mathcal{F}$ be the resulting set of spheres. Compute the arrangement $\mathcal{A}(\mathcal{F})$, and place a point inside each face of this arrangement. Let Q be the resulting set of points. Compute the point $c \in Q$ realizing $\max_{f \in Q} \nabla(f, C)$, and return it as the desired new center.

To show the correctness, consider the (open) face F of $\mathcal{A}(C)$, that contains c^*, where c^* is the optimal center to be added. Let f be any point of Q in F. Let

$$\nabla(f,p) = \rho(f \cup C, p) - \rho(C,p).$$

Define $\nabla(c^*, p)$ similarly. Clearly, we have $\nabla(f, C) = \sum_{p \in P} \nabla(f,p)$. Let P_1 be all the points p of P such that $\nabla(c^*,p) \leq \nabla(f,p) + \varepsilon\Delta/(4n)$. Similarly, let P_2 be all the points p of P, such that

$$\nabla(c^*,p) > \nabla(f,p) + \varepsilon\Delta/(4n).$$

For any point $p \in P_2$, by the choice of N, there exists an index i, such that $\ell(i) \leq \nabla(c^*,p) < \ell(i+1)$. By the choice of f from the arrangement, we have that $\nabla(f,p) \geq \ell(i)$, which in turn implies that

$$\nabla(f,p) \leq \nabla(c^*,p) < (1+\varepsilon/4)\nabla(f,p) \implies \nabla(f,p) \geq (1-\varepsilon/2)\nabla(c^*,p).$$

We thus have the following

$$\nabla(f,C) = \sum_{p \in P} \nabla(f,p) = \sum_{p \in P_1} \nabla(f,p) + \sum_{p \in P_2} \nabla(f,p)$$

$$\geq \sum_{p\in P_1} \big(\nabla(c^*,p) - \frac{\varepsilon\Delta}{4n}\big) + \sum_{p\in P_2} (1-\varepsilon/2)\nabla(c^*,p)$$
$$\geq (1-\varepsilon/2)\sum_{p\in P} \nabla(c^*,p) - \frac{\varepsilon\Delta}{4} \geq (1-\varepsilon)\nabla(c^*,C).$$

The runtime follows from the observation that the number of faces in the arrangement is $(n/\varepsilon)^{O(d)}$. ◀

▶ **Lemma 6.** *Given a set P of n points in the plane, and a parameter k, one can compute in polynomial time (i.e., $n^{O(d)}$) a constant approximation to $\rho_{\mathsf{O}} = \mathrm{opt}_k(P)$.*

Proof. Follows by using Lemma 5 in a greedy fashion k times, with $\varepsilon = 0.1$, to get a set of k centers. The quality of approximation readily follows from known results about submodularity [18]. Indeed, let $v_i = \rho(C_i, P)$ be the service provided by the first i centers computed. By submodularity, and the quality guarantee of Lemma 5, we have that $\nabla(c_i, C_{i-1}) \geq (1-\varepsilon)(\rho_{\mathsf{O}} - v_{i-1})/k$. In particular, setting $\Delta_0 = \rho_{\mathsf{O}}$, and $\Delta_i = \rho_{\mathsf{O}} - v_{i-1}$, we have that $\Delta_i \leq (1-(1-\varepsilon)/k)\Delta_{i-1}$. As such, $\Delta_k \leq \exp(-k(1-\varepsilon)/k)\Delta_0 = \rho_{\mathsf{O}}/e^{\varepsilon-1} \leq \rho_{\mathsf{O}}/2$. Namely, we have $v_k \geq \rho_{\mathsf{O}}/2$, as desired. ◀

For two sets of points S and C we define $\nabla(S,C) = \rho(C+S,P) - \rho(C-S,P)$.

▶ **Lemma 7.** *Let P be a set of n points in $\mathbb{R}^d$, and let C be a set of k centers. Given an integer $t \geq 1$, a parameter $\varepsilon \in (0,1)$, in $(n/\varepsilon)^{O(dt)}$ time, one can $(1-\varepsilon)$-approximate the set $S \subset \mathbb{R}^d$ with $|S| = t$ that maximizes the marginal value in $(n/\varepsilon)^{O(td)}$ time. Formally, we have $\nabla(S,C) \geq (1-\varepsilon)\max_{F\subset\mathbb{R}^d,\ |F|=t} \nabla(F,C)$.*

Proof. Consider the optimal set F of size t. Denote it by S^*. Compute the same arrangement as in the proof of Lemma 5. Let $F_1, \ldots, F_t$ be the faces of $\mathcal{A}(C)$ that contain the t points of S^*. Pick an arbitrary point from each F_i and let S be the resulting point set of size t. Define

$$\nabla(S,p) = \rho(S\cup C, p) - \rho(C,p).$$

and similarly $\nabla(S^*,p)$.

As in the proof of Lemma 5, let P_1 be all the points of P such that $\nabla(S^*,p) \leq \nabla(S,p) + \varepsilon\Delta/4n$ and P_2 be all the points of P such that

$$\nabla(S^*,p) > \nabla(S,p) + \varepsilon\Delta/4n.$$

We also conclude that $\nabla(S,p) \geq (1-\varepsilon/2)\nabla(S^*,p)$ for all $p \in P_2$. We get

$$\nabla(S,C) = \sum_{p\in P} \nabla(S,p) = \sum_{p\in P_1} \nabla(S,p) + \sum_{p\in P_2} \nabla(S,p)$$
$$\geq \sum_{p\in P_1} \big(\nabla(S^*,p) - \frac{\varepsilon\Delta}{4n}\big) + \sum_{p\in P_2} (1-\varepsilon/2)\nabla(S^*,p)$$
$$\geq (1-\varepsilon/2)\sum_{p\in P} \nabla(S^*,p) - \frac{\varepsilon\Delta}{4} \geq (1-\varepsilon)\nabla(S^*,C).$$

The runtime follows from the observation that the number of faces in the arrangement is $(n/\varepsilon)^{O(d)}$ and that it is sufficient to consider subsets of size t of the faces. ◀

4 The local search algorithm

The algorithm starts with a constant approximation, using the algorithm of Lemma 6. Next, the algorithm performs local exchanges, as long as it can find a local exchange that is sufficiently profitable.

Specifically, let $\alpha = O(1/\varepsilon^d)$, and let $\xi = O(\alpha/\varepsilon) = O(1/\varepsilon^{d+1})$. Assume that one can "quickly" check given a set of k centers C, whether there is a local exchange of size ξ, such that the resulting set of centers provides service $(1+\varepsilon^2/(16k))\rho_{\mathrm{curr}}$, where ρ_{curr} is the service of the current solution. To this end, the algorithm considers at most k^ξ possible subsets of the current set of centers that might be dropped, and for each such subset, one can apply Lemma 7, to compute (approximately) the best possible centers to add. If all such subsets do not provide an improvement, the algorithm stops.

Running time analysis

The algorithm starts with a constant approximation. As such, there could be at most $O(k/\varepsilon^2)$ local exchanges before the algorithm must terminate. Finding a single such exchange requires applying Lemma 7 k^ξ times. Lemma 7 is invoked with $t = \xi$. The resulting running time is $k^\xi(n/\varepsilon)^{O(d\xi)} = (n/\varepsilon)^{O(d/\varepsilon^{d+1})}$, where we remember that $k \le n$.

5 Correctness of the local search algorithm

Here we show that if the local algorithm has reached a local optimum, then it reached a solution that is a good approximation to the optimal solution.

▶ Remark 8. In the following, we simplify the analysis at some points, by assuming that the local solution takes an exchange if it provides any improvement (the algorithm, however, takes an exchange only if it is a significant improvement). Getting rid of the assumption and modifying the analysis is straightforward, but tedious.

5.1 Notations

Let L and O be the local and optimal set of k centers. Let $\mathcal{U} = \mathsf{L} \cup \mathsf{O}$. Assign a point of $\mathcal{U}$ color $+1$ if it is in O and -1 if it is in L (for the sake of simplicity of exposition assume no point belong to both sets). Let γ be a sufficiently large constant. For $\delta = \varepsilon/\gamma$ and $\alpha = O(1/\delta^{d+1})$, compute a α-division $\mathcal{D} = \{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$ of $\mathsf{L} \cup \mathsf{O}$, using Theorem 4.

Let $\mathsf{L}_i = B_i \cap \mathsf{L}$, $\overline{\mathsf{L}}_i = \mathsf{L}_i \cup \partial_i$, $\mathsf{O}_i = B_i \cap \mathsf{O}$, $\overline{\mathsf{O}}_i = \mathsf{O}_i \cup \partial_i$, $\ell_i = |\mathsf{L}_i|$, and $o_i = |\mathsf{O}_i|$, for all i. Let $\overline{\mathsf{L}} = \bigcup_i \overline{\mathsf{L}}_i$, and $\overline{\mathsf{O}} = \bigcup_i \overline{\mathsf{O}}_i$. Let $\partial = \bigcup_i \partial_i$. By construction, we have that $\sum_i |\partial_i| \le \delta 2k \le \varepsilon k/4$ if γ is a sufficiently large constant.

5.2 Submodularity implies slow degradation

The following is a well known implication of submodularity. We include the proof for the sake of completeness.

▶ **Lemma 9.** *Let C be a set of k centers. Then, for any $t \le k$, there exists a subset $C' \subseteq C$ of size t, such that $\rho(C') \ge \frac{t}{k}\rho(C)$, where $\rho(C) = \rho(C, P)$.*

Proof. Let $C_0 = C$. In the ith iteration, we greedily remove the point of C_{i-1} that is minimizing the marginal value. Formally,

$$f_i = \arg\min_{c \in C_{i-1}} \nabla(c, C_{i-1} - c),$$

and $C_i = C_{i-1} - f_i$. By submodularity, we have that $\nabla(f_i, C_{i-1} - f_i) \leq \rho(C_{i-1})/|C_{i-1}|$. As such, we have

$$\rho(C_i) = \rho(C_{i-1}) - \nabla(f_i, C_{i-1} - f_i) \geq \left(1 - \frac{1}{k-i+1}\right)\rho(C_{i-1}) = \frac{k-i}{k-i+1}\rho(C_{i-1})$$
$$\geq \frac{k-i}{k-i+1} \cdot \frac{k-i+1}{k-i+1+1} \cdots \frac{k-1}{k}\rho(C_0) = \frac{k-i}{k}\rho(C).$$

The claim now readily follows by taking the set C_{k-t}. ◀

5.3 Boundary vertices are not profitable

First, we argue that adding the boundary points, does not increase the profit/service significantly, for either the local or optimal solutions.

▶ **Lemma 10.** *$\rho(\overline{\mathsf{L}}) \leq (1+\varepsilon/4)\rho(\mathsf{L})$ and $\rho(\overline{\mathsf{O}}) \leq (1+\varepsilon/4)\rho(\mathsf{O})$.*

Proof. Let $p = |\partial|$. Consider a point $c \in \partial$, and observe that $\nabla(c, \mathsf{L}) \leq \rho(\mathsf{L})/k$. This is a standard consequence of submodularity and greediness/local optimality. To see that, order the centers of $\mathsf{L} = \{c_1, \ldots, c_k\}$ in an arbitrary order. Let $\nabla_i = \nabla(c_i, \{c_1, \ldots, c_{i-1}\}) \geq 0$, for $i = 1, \ldots, k$, and observe that $\rho(\mathsf{L}) = \sum_{i=1}^{k} \nabla_i$. As such, there exists an index i, such that $\nabla_i \leq \rho(\mathsf{L})/k$. By submodularity, we have that $\nabla(c_i, \mathsf{L} - c_i) \leq \nabla(c_i, \{c_1, \ldots, c_{i-1}\}) = \nabla_i \leq \rho(\mathsf{L})/k$, and $\nabla(c, \mathsf{L} - c_i) \geq \nabla(c, \mathsf{L})$. Assume, for the sake of contradiction, that $\nabla(c, \mathsf{L}) > \rho(\mathsf{L})/k$. We have that

$$\rho(\mathsf{L} - c_i + c) = \rho(\mathsf{L}) - \nabla(c_i, \mathsf{L} - c_i) + \nabla(c, \mathsf{L} - c_i) \geq \rho(\mathsf{L}) - \frac{\rho(\mathsf{L})}{k} + \nabla(c, \mathsf{L})$$
$$> \rho(\mathsf{L}) - \frac{\rho(\mathsf{L})}{k} + \frac{\rho(\mathsf{L})}{k} = \rho(\mathsf{L}).$$

But the local search algorithm considered this swap, which means that $\mathsf{L} - c_i + c$ can not be more profitable than the local solution. A contradiction (see Remark 8).

Setting $\partial = \{f_1, \ldots, f_p\}$, we have

$$\rho(\overline{\mathsf{L}}) = \rho(\mathsf{L}) + \sum_{i=1}^{p} \nabla(f_i, \mathsf{L} + f_1 + \cdots + f_{i-1}) \leq \rho(\mathsf{L}) + \sum_{i=1}^{p} \nabla(f_i, \mathsf{L}) \leq$$
$$\rho(\mathsf{L}) + p\rho(\mathsf{L})/k \leq (1+\varepsilon/4)\rho(\mathsf{L}),$$

since $p = |\partial| \leq \varepsilon k/4$.

The second claim follows by a similar argument. ◀

5.4 If there is a gap, then there is a swap

The contribution of the clusters L_i and O_i is

$$\nabla\mathsf{L}_i = \nabla\big(\mathsf{L}_i, \overline{\mathsf{L}} \setminus \mathsf{L}_i\big). \qquad \text{and} \qquad \nabla\mathsf{O}_i = \nabla\big(\mathsf{O}_i, \overline{\mathsf{O}} \setminus \mathsf{O}_i\big), \tag{5.1}$$

respectively. Notice that, because of the separation property, the points in P that their coverage change when we move from $\overline{\mathsf{L}} \setminus \mathsf{L}_i$ to $\overline{\mathsf{L}}$, are points that are served by $\partial_i \subseteq \overline{\mathsf{L}} \setminus \mathsf{L}_i$ (same holds for $\overline{\mathsf{O}} \setminus \mathsf{O}_i$ and $\overline{\mathsf{O}}$).

This implies that

$$\nabla(\mathsf{L}, \partial) = \sum_i \nabla\mathsf{L}_i \qquad \text{and} \qquad \nabla(\mathsf{O}, \partial) = \sum_i \nabla\mathsf{O}_i.$$

In the following, we assume that $\rho(\mathsf{L}) < (1-\varepsilon)\rho(\mathsf{O})$. By Lemma 10 this implies that $\rho(\overline{\mathsf{L}}) \leq (1+\varepsilon/4)\rho(\mathsf{L}) < (1+\varepsilon/4)(1-\varepsilon)\rho(\mathsf{O}) \leq (1-\varepsilon/2)\rho(\mathsf{O}) \leq (1-\varepsilon/2)\rho(\overline{\mathsf{O}})$. As such, we have

$$\begin{aligned}
\rho(\overline{\mathsf{L}}) < (1-\varepsilon/2)\rho(\overline{\mathsf{O}}) &\implies \rho(\partial) + \nabla(\mathsf{L},\partial) < (1-\varepsilon/2)(\rho(\partial) + \nabla(\mathsf{O},\partial)) \\
&\implies \nabla(\mathsf{L},\partial) < (1-\varepsilon/2)\nabla(\mathsf{O},\partial) - (\varepsilon/2)\rho(\partial) \\
&\implies \nabla(\mathsf{L},\partial) < (1-\varepsilon/2)\nabla(\mathsf{O},\partial) \\
&\implies (\varepsilon/2)\nabla(\mathsf{O},\partial) < \sum_i (\nabla \mathsf{O}_i - \nabla \mathsf{L}_i).
\end{aligned}$$

By averaging, this implies that there exists an index t, such that

$$\nabla \mathsf{O}_t - \nabla \mathsf{L}_t > \frac{\varepsilon}{2k}\nabla(\mathsf{O},\partial) > \frac{\varepsilon}{2k}\big(\nabla(\mathsf{O},\partial) - \nabla(\mathsf{L},\partial)\big) \tag{5.2}$$

$$= \frac{\varepsilon}{2k}\big(\rho(\overline{\mathsf{O}}) - \rho(\overline{\mathsf{L}})\big) \geq \frac{\varepsilon}{2k}\cdot\frac{\varepsilon}{2}\rho(\overline{\mathsf{O}}) \geq \frac{\varepsilon^2}{4k}\rho(\overline{\mathsf{O}}), \tag{5.3}$$

where in the second to last inequality we use that $\rho(\overline{\mathsf{L}}) < (1-\varepsilon/2)\rho(\overline{\mathsf{O}})$. Namely, there is a batch where the local and optimal solution differ significantly.

5.4.1 An unlikely scenario

Assume that $|\mathsf{L}_t| \geq |\mathsf{O}_t| + |\partial_t|$. We then have that

$$\rho(\mathsf{L} + \partial_t - \mathsf{L}_t + \mathsf{O}_t) = \rho(\mathsf{L} + \partial_t) - \nabla \mathsf{L}_t + \nabla \mathsf{O}_t \geq \rho(\mathsf{L}) + \frac{\varepsilon^2}{4k}\rho(\overline{\mathsf{O}}).$$

But this is impossible, since the local search algorithm would have performed the exchange $\mathsf{L} + \partial_t - \mathsf{L}_t + \mathsf{O}_t$, since $|\partial_t| + |\mathsf{L}_t| + |\mathsf{O}_t|$ is smaller than the size of exchanges considered by the algorithm.

5.4.2 The general scenario

▶ **Lemma 11.** *There exists a subset $Y \subseteq \mathsf{O}_t$, such that $|\mathsf{L}_t| \geq |Y| + |\partial_t|$, and*

$$\nabla\big(Y, \overline{\mathsf{O}} \setminus \mathsf{O}_t\big) \geq \nabla \mathsf{L}_t + \frac{\varepsilon^2}{8k}\rho(\overline{\mathsf{O}}),$$

see Eq. (5.1).

Proof. We have that $(\varepsilon/2)\nabla(\mathsf{O},\partial) < \sum_i (\nabla \mathsf{O}_i - \nabla \mathsf{L}_i)$. Subtracting $(\varepsilon/8)\nabla(\mathsf{O},\partial)$ from both sides implies that $(\varepsilon/4)\nabla(\mathsf{O},\partial) < \sum_i ((1-\varepsilon/8)\nabla \mathsf{O}_i - \nabla \mathsf{L}_i)$. This in turn implies that there exists t such that $(1-\varepsilon/8)\nabla \mathsf{O}_t - \nabla \mathsf{L}_t > (\varepsilon/(4k))\nabla(\mathsf{O},\partial)$. Arguing, as above, we have that $(1-\varepsilon/8)\nabla \mathsf{O}_t - \nabla \mathsf{L}_t > (\varepsilon^2/(8k))\rho(\overline{\mathsf{O}})$.

Consider the following (submodular) function

$$f(X) = \nabla\big(X, \overline{\mathsf{O}} \setminus \mathsf{O}_t\big),$$

By Lemma 9 (or more precisely arguing as in this lemma), we have that there exists a set $Y \subset \mathsf{O}_t$, such that $|Y| = (1-\varepsilon/8)\,|\mathsf{O}_t|$ and $f(Y) \geq (1-\varepsilon/8)f(\mathsf{O}_t) = (1-\varepsilon/8)\nabla \mathsf{O}_t$. As such, we have that

$$\nabla\big(Y, \overline{\mathsf{O}} \setminus \mathsf{O}_t\big) \geq (1-\varepsilon/8)\nabla \mathsf{O}_t \geq \nabla \mathsf{L}_t + (\varepsilon^2/8k)\rho(\overline{\mathsf{O}}).$$

As for the size of Y. Observe that by Theorem 4, we have $|\partial_i| \leq \frac{\varepsilon}{\gamma}(|\mathsf{O}_i| + |\mathsf{L}_i|)$ and $\big||\mathsf{O}_i| - |\mathsf{L}_i|\big| \leq \frac{\varepsilon}{\gamma}(|\mathsf{O}_i| + |\mathsf{L}_i|)$. This readily implies that $|\mathsf{O}_i| \leq (1 + 4\varepsilon/\gamma)\,|\mathsf{L}_i|$, and $|\mathsf{L}_i| \leq (1 + 4\varepsilon/\gamma)\,|\mathsf{O}_i|$, if γ is sufficiently large. As such, we have that

$$|\mathsf{L}_t| \geq \frac{|\mathsf{O}_t|}{1 + 4\varepsilon/\gamma} \geq (1 - 4\varepsilon/\gamma)\,|\mathsf{O}_t| = (1 - \varepsilon/8)\,|\mathsf{O}_t| + (\varepsilon/8 - 4\varepsilon/\gamma)\,|\mathsf{O}_t|$$
$$\geq |Y| + \frac{\varepsilon}{16}\,|\mathsf{O}_t| \geq |Y| + |\partial_i|\,,$$

if $\gamma \geq 64$. ◀

▶ **Lemma 12.** *The local search algorithm computes a $(1-\varepsilon)$-approximation to the optimal solution.*

Proof. If not, then, arguing as above, there must be a batch for which there is an exchange with profit at least $(\varepsilon^2/4k)\rho\big(\overline{\mathsf{O}}\big)$ (see Eq. (5.3)). By Lemma 11, we can shrink the optimal batch O_t, such that the exchange becomes feasible, and is still profitable (the profit becomes $(\varepsilon^2/8k)\rho\big(\overline{\mathsf{O}}\big)$). But that is impossible, since by arguing as above (i.e., the unlikely scenario), we have that this swap would result in a better local solution, and the exchange is sufficiently small to have been considered. Specifically, the local search algorithm uses Lemma 7, say with $\varepsilon = 1/2$, ensures that the local search algorithm would find an exchange with half this value, and would take it. A contradiction. ◀

5.5 The result

▶ **Theorem 13.** *Let P be a set of n points in $\mathbb{R}^d$, let $\varepsilon \in (0,1)$ be a parameter, let $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ be a service function, and let $k \leq n$ be an integer parameter. One can compute, in $(n/\varepsilon)^{O(d/\varepsilon^{d+1})}$ time, a set of k centers C, such that $\rho(C,P) \geq (1-\varepsilon)\mathrm{opt}_k(P)$, where $\mathrm{opt}_k(P)$ denotes the optimal solution using k centers.*

6 Discussion

We presented an algorithm that runs in polynomial time for any constant $\varepsilon > 0$ and any constant dimension d and achieves a $(1-\varepsilon)$-approximation. The dependency on the dimension d is doubly exponential, however. A natural question is whether the dependency on the dimension d in the runtime can be improved. Perhaps by considering some special cases of the problem, for more specific service function, such as $\varphi(\ell) = \frac{1}{1+\ell}$ or $\varphi(\ell) = \frac{1}{1+\ell^2}$ (i.e., the service quality drops roughly linearly or quadratically with the distance).

Our algorithm finds a subset $C \subseteq \mathbb{R}^d$ of size k that approximately maximizes the objective function. A close variant of the problem asks to find a subset $C \subseteq \mathbb{R}^d$ of size k that has an additional constraint that $C \subseteq P$. Can we obtain an algorithm for this problem with the same asymptotic runtime for $d > 2$? The main difficulty is that we would need a variant of Theorem 4 with an additional property $\partial_i \subseteq P$ for all i but such a division does not exist even for $d = 3$. (For $d = 2$ using planar graph divisions on the Voronoi diagram directly implies the desired result.)

Finally, one can ask a similar question about clustering in graphs. Specifically, given a graph on n vertices P, we would like to select k vertices C that approximately maximizes $\sum_{p \in P} \min_{c \in C} \varphi\big(\mathsf{d}(p,c)\big)$, where $\mathsf{d}(p,c)$ is the shortest-path distance from p to c. Can we achieve a polynomial time algorithm for arbitrary small constant $\varepsilon > 0$ if the graph is planar or come from some other class of graphs?

References

1 Arturs Backurs, Moses Charikar, Piotr Indyk, and Paris Siminelakis. Efficient density evaluation for smooth kernels. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 615–626. IEEE, 2018.

2 Vijay V. S. P. Bhattiprolu and Sariel Har-Peled. Separating a Voronoi diagram via local search. In Sándor P. Fekete and Anna Lubiw, editors, *Proc. 32nd Int. Annu. Sympos. Comput. Geom.* (SoCG), volume 51 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.SoCG.2016.18.

3 Steven Chaplick, Minati De, Alexander Ravsky, and Joachim Spoerhase. Approximation schemes for geometric coverage problems. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *Proc. 27th Annu. Euro. Sympos. Alg.* (ESA), volume 112 of *LIPIcs*, pages 17:1–17:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.17.

4 Moses Charikar and Paris Siminelakis. Hashing-based-estimators for kernel density in high dimensions. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1032–1043. IEEE, 2017.

5 Vincent Cohen-Addad. A fast approximation scheme for low-dimensional k-means. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 430–440. Society for Industrial and Applied Mathematics, 2018.

6 Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-Linear Time Approximation Schemes for Clustering in Doubling Metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 2019.

7 Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. *SIAM J. Comput.*, 48(2):644–667, 2019. doi:10.1137/17M112717X.

8 Vincent Cohen-Addad and Claire Mathieu. Effectiveness of local search for geometric optimization. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, volume 34 of *LIPIcs*, pages 329–343. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.SOCG.2015.329.

9 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. doi:10.1007/978-3-540-77974-2.

10 R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, New York, 2nd edition, 2001.

11 Zachary Friggstad, Kamyar Khodamoradi, Mohsen Rezapour, and Mohammad R Salavatipour. Approximation schemes for clustering with outliers. *ACM Transactions on Algorithms (TALG)*, 15(2):26, 2019.

12 Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *SIAM Journal on Computing*, 48(2):452–480, 2019.

13 Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. In *ICCV*, volume 3, page 456, 2003.

14 Leslie Greengard and John Strain. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

15 Kai Jin, Jian Li, Haitao Wang, Bowei Zhang, and Ningye Zhang. Near-linear time approximation schemes for geometric maximum coverage. *Theoretical Computer Science*, 725:64–78, 2018. doi:10.1016/j.tcs.2017.11.026.

16 Kiyohito Nagano, Yoshinobu Kawahara, and Satoru Iwata. Minimum average cost clustering. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Neural Info. Proc. Sys.* (NIPS), pages 1759–1767. Curran Associates, Inc., 2010. URL: http://papers.nips.cc/paper/4106-minimum-average-cost-clustering.

17 Kai Wei, Rishabh K. Iyer, Shengjie Wang, Wenruo Bai, and Jeff A. Bilmes. Mixed robust/average submodular partitioning: Fast algorithms, guarantees, and applications. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Neural Info. Proc. Sys.* (NIPS), pages 2233–2241, 2015. URL: `http://papers.nips.cc/paper/5706-mixed-robustaverage-submodular-partitioning-fast-algorithms-guarantees-and-applications`.

18 L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982. `doi:10.1007/BF02579435`.

A Balanced Voronoi division

We need the following variant of a result of Bhattiprolu and Har-Peled [2].

▶ **Theorem 14.** *Let P be a set of n' points in $\mathbb{R}^d$, where every point has a positive integer weight, such that the total weight of the points is n, and let α be parameter. Furthermore, assume that no point has weight that exceeds α. Then, one can compute, in expected $O(n)$ time, a ball , and a set Z that lies on the boundary of , such that*

(i) $|Z| \leq \mathsf{c}_1 \alpha^{1-1/d}$,
(ii) *the total weight of the points of P inside is at least α and at most $\mathsf{c}_2\alpha$,*
(iii) *Z is a Voronoi separator of the points of P inside from the points of P outside .*

Here $\mathsf{c}_1, \mathsf{c}_2 > 0$ are constants that depends only on the dimension d.

The points of the separator Z are guards.

A.1 A division using the above separator

A.1.1 Algorithm

We start with a set P of n points, and a parameter α. The idea is to repeatedly extract a set of weight (roughly) α from the point set, separate it, remove it, and put the set of guards associated with it back into the set.

To this end, let α be a parameter, such that

$$\mathsf{c}_1 \alpha^{1-1/d} < \alpha/8 \iff 8\mathsf{c}_1 < \alpha^{1/d} \iff \alpha > (8\mathsf{c}_1)^d,$$

where c_1 is the constant from Theorem 14.

For an unweighted set of points X and a real number $\tau > 0$, let $\tau * X$ denote the set of points, where every points has weight τ.

The algorithm for constructing the division is the following:

1. $P_0 \leftarrow P$. Initially all the points in P_0 have weight 1.
2. $i \leftarrow 1$.
3. While P_{i-1} has total weight larger than α do:
   3.1 $(\text{ball}_i, Z_i) \leftarrow$ ball and separator computed by Theorem 14 for P_{i-1} with parameter α.
   3.2 $I_i \leftarrow P_{i-1} \cap \text{ball}_i$. // All points inside ball to be removed
   3.3 $G_i \leftarrow I_i \setminus P$. // The old guards in the ball
   3.4 $B_i = P \cap I_i$ // The batch of original points
   3.5 $P_i = (P_{i-1} \setminus \text{ball}_i) \cup (\tau_i * Z_i)$, where $\tau_i = \lceil (\alpha/4)/|Z_i| \rceil$.
   3.6 $\partial_i = Z_i \cup G_i$ // The set of guards for the batch B_i
   3.7 $i \leftarrow i + 1$.
4. $m \leftarrow i$
5. $B_m = P_{m-1} \cap P$, and $\partial_m = P_{m-1} \setminus B_m$.
6. Return $\mathcal{D} = \{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$.

A.1.2 Analysis

▶ **Lemma 15.** *Consider the Voronoi diagram of $\mathcal{V}(P \cup \partial_i)$. There is no common boundary in this Voronoi diagram between a cell of a point of B_i and a cell of a point of $P \setminus B_i$.*

Proof. Consider a point p that is in equal distance to a point $f \in B_i$, and a point $g \in P \setminus B_i$, and furthermore, all other points of $P \setminus \{f, g\}$ are strictly further away from p.

The claim is that $\mathsf{d}(p, \partial_i) < \|p - f\| = \|p - g\|$. Namely, the region of common boundary between f and g in $\mathcal{V}(P)$ is completely covered by cells of ∂_i in $\mathcal{V}(P \cup \partial_i)$.

If $g \in P_i$, then Z_i separates (in the Voronoi interpretation) $f \in B_i \subseteq I_i$ from all the points of $P_i \cap P \ni g$, which implies the claim.

As such, it must be that $g \in B_j$, for some $j < i$. Namely, there is a guard $g_j \in Z_j$, that separates g from f, and its cell contains p. That is $\|p - g_j\| < \|p - f\|$, and $g_j \in P_j$. If $g_j \in \partial_i$ then the claim holds.

Otherwise, we apply the same argument again, this time to g_j and f. Indeed, g_j was removed (from P_k) in some iteration k, such that $j < k < i$. Namely $g_j \in {}_k$, and $f \notin {}_k$. The point p is closer to g_j then to f. If $p \in {}_k$ then there is a guard $g_k \in Z_k$ that is closer to p than f, by the separation property. Otherwise, it is easy to verify that the Voronoi cells of the guards of Z_k in $\mathcal{V}(P_{k-1} \cup Z_k)$ cover completely the portion of the Voronoi cells of points in $P_{k-1} \cap {}_k$ outside ${}_k$, in the Voronoi diagram $\mathcal{V}(P_{k-1})$. This readily implies that there is a closer guard $g_k \in Z_k$ to p than g_j. In either case, we continue the argument inductively on (g_k, f).

By finiteness, it follows that there must be a guard $g' \in \partial_i$ that is closer to p than f, which implies the claim. ◀

▶ **Observation 16.** *(A) For all i, we have $\tau_i = \left\lceil \frac{\alpha/4}{|Z_i|} \right\rceil \geq \frac{\alpha/4}{|Z_i|} \geq \frac{\alpha/4}{\mathsf{c}_1 \alpha^{1-1/d}} \geq \frac{\alpha^{1/d}}{4\mathsf{c}_1}$.*

(B) As such, for all i, I_i contains at most $\mathsf{c}_2 \alpha / \min_j \alpha_j = O(\alpha^{1-1/d})$ points that are not in P. That is, we have $|I_i \setminus B_i| = O(\alpha^{1-1/d})$.

(C) It follows that $|\partial_i| = |I_i \setminus B_i| + |Z_i| = O(\alpha^{1-1/d})$.

▶ **Lemma 17.** *Given a set P of n points in $\mathbb{R}^d$, and a parameter α, one can compute in polynomial time, a division $\mathcal{D} = \{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$, such that the following holds:*

(A) *$\bigcup B_i = P$, and the clusters $B_1, \ldots, B_m$ are disjoint.*

(B) *$m = O(n/\alpha)$.*

(C) *For all i, we have the following properties:*

- **(C.i)** *the set ∂_i separates B_i from $P \setminus B_i$.*
- **(C.ii)** *$|B_i| = O(\alpha)$.*
- **(C.iii)** *$|\partial_i| = O(\alpha^{1-1/d})$.*

(D) *For $\partial = \bigcup_i \partial_i$, we have that $|\partial| = O(n/\alpha^{1/d})$.*

Furthermore, one can modify the above construction, so that Cii is replaced by $|B_i| = \Theta(\alpha)$.

Proof. For the bound on number of clusters, observe that every iteration of the algorithm reduces the weight of the working set P_i by at least $\alpha/2$ – indeed, the weight of B_i is at least α, and the total weight of points of Z_i (after multiplying their weight by τ_i) is at most $\alpha/2$. Thus implying the claim.

All the other claims are either proved above, or readily follows from the algorithm description.

The modification of Cii follows by observing that we can merge clusters, and there are $\Theta(n/\alpha)$ clusters with $\Omega(\alpha)$ points of P, by averaging. As such, one can merge $O(1)$ clusters

that have $o(\alpha)$ points of P into a cluster that has $\Omega(\alpha)$ points of P, thus implying the modified claim. ◀

▶ **Theorem 4.** *Given a set P of n points in $\mathbb{R}^d$, parameters $\delta \in (0,1)$ and $\alpha = \Omega(1/\delta^{d+1})$, and a balanced coloring χ of P, one can compute in polynomial time, a Voronoi α-division $\mathcal{D} = \{(B_1, \partial_1), \ldots, (B_m, \partial_m)\}$, such that the following holds:*

(A) $\bigcup B_i = P$, *and the batches $B_1, \ldots, B_m$ are disjoint.*

(B) $m = O(n/\alpha)$.

(C) *For all i, we have the following properties:*

- **(C.i)** *the set ∂_i Voronoi separates B_i from $P \setminus B_i$.*
- **(C.ii)** $(1-\delta)\alpha \leq |B_i| \leq \alpha$ *(except for the last batch, which might be of size at least $(1-\delta)\alpha$, and at most size 2α).*
- **(C.iii)** $|\partial_i| \leq \delta\,|B_i|$.
- **(C.iv)** $|\chi(B_i)| \leq \delta\,|B_i|$.

Proof. We compute a division of P using Lemma 17, with parameter $\alpha' = O(\delta\alpha) = \Omega(1/\delta^d)$, such that (i) the maximum size of a batch is strictly smaller than $\delta\alpha/2$, and (ii) $\mathsf{c}_4(\alpha')^{1-1/d} < \delta\alpha'$, where c_4 is some prespecified constant. Let $\mathcal{D}' = \{(B'_1, \partial'_1), \ldots, (B'_\tau, \partial'_\tau)\}$, be the resulting division. Let $\Delta_i = \chi(\partial'_i)$, and observe that $\sum_i \Delta_i = \chi(P) = 0$. There is a permutation π of the batches, such that for any prefix j, we have $|\sum_{i=1}^{j} \Delta_{\pi(i)}| \leq \max_i |B_i| \leq \delta\alpha/2 = D$. This follows readily by reordering the summation, such that one adds batches with positive (resp., negative) balance if the current prefix sum is negative (resp., positive), and repeating this till all the terms are used[2].

We break the permutation π into minimum number of consecutive intervals, such that total size of batches in each interval is at least $(1-\delta)\alpha$. Merging the last two interval if needed to comply with the desired property. The batch formed by a union of an interval can have discrepancy at most $2D = 2(\delta\alpha/2) = \delta\alpha$, as desired.

All the other properties follows readily by observing that merging batches, results in valid batches, as far as separation. ◀

[2] This is the same idea that is used in the Riemann rearrangement theorem.