ALGORITHMS ON CLUSTERING, ORIENTEERING, AND CONFLICT-FREE COLORING

BY

KE CHEN

B.Eng., Huazhong University of Science and Technology, 1999
M.Eng., Huazhong University of Science and Technology, 2001

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2007

Urbana, Illinois

Doctoral Committee:

Associate Professor Sariel Har-Peled, Chair
Associate Professor Chandra Chekuri
Associate Professor Jeff Erickson
Professor Lenny Pitt

Abstract

We present algorithms for three geometric problems — clustering, orienteering, and conflict-free coloring.

In the first part, we obtain small *coresets* for k-median and k-means clustering in general metric spaces and in Euclidean spaces. In $\mathbb{R}^d$, these coresets have size that depend polynomially on d. This leads to more efficient approximation algorithms for k-median and k-means clustering. We use those coresets to maintain a $(1+\varepsilon)$-approximate k-median and k-means clustering of a stream of points in $\mathbb{R}^d$, using $O(dk^2\varepsilon^{-2}\log^8 n)$ space. These are the first streaming algorithms, for those problems, that have space complexity with *polynomial* dependency on the dimension.

We next study the *k-median clustering with outliers* problem. Here, given a finite point set in a metric space and parameters k and m, we want to remove m points (called outliers), such that the cost of the optimal k-median clustering of the remaining points is minimized. We present the first polynomial time constant factor approximation algorithm for this problem.

In the second part, we consider the *rooted orienteering* problem: Given a set P of n points in the plane, a starting point $r \in P$, and a length constraint $\mathcal{B}$, one needs to find a path starting from r that visits as many points of P as possible and of length not exceeding $\mathcal{B}$. We present the first polynomial time approximation scheme (PTAS) for this problem. The scheme also works in higher (fixed) dimensions.

In the last part, we present randomized algorithms for *online conflict-free coloring* of points in the plane, with respect to intervals, halfplanes, congruent disks, and nearly-equal axis-parallel rectangles. In all these cases, the coloring algorithms use $O(\log n)$ colors, with high probability. We also present the first efficient deterministic algorithm for the CF coloring of points in the plane with respect to nearly-equal axis-parallel rectangles, using $O(\log^{12} n)$ colors.

Acknowledgments

I would like to thank the many people who helped and supported me during my time at UIUC.

First and foremost, I am grateful to my advisor, Sariel Har-Peled, for his support and enthusiastic supervision, from insightful comments on my research to tiny punctuation corrections in this thesis. I particularly appreciate his patience in going through numerous iterations to help me make better presentations of my ideas. His attitude toward research and life in general have been truly enlightening. Thanks, Sariel! I have learned a great deal from you.

I thank Haim Kaplan and Micha Sharir for valuable discussions on the problems studied in this thesis.

Many other faculty members at UIUC contributed to my success. Thanks to Chandra Chekuri and Jeff Erickson for always being available and for their useful advices on research and my career. Thanks to Lenny Pitt, Margaret Fleck, and Madhusudan Parthasarathy for demonstrating excellence in teaching and providing me with ample opportunities to benefit from interacting with students when I was a TA for their classes.

I enjoyed the interactions with many graduate students at UIUC, and I thank my office mates for being great colleagues.

Last but not least, I would like to express my love and gratitude to my parents and my brother for their love and support throughout all these years. I thank my wife, Shuang Liang, for her love and patience. This work is dedicated to them.

Table of Contents

List of Figures

Part I

Clustering

Chapter 1

Introduction

Clustering is the process of classifying a set of objects into groups such that objects of each group are similar. It is an important problem in computer science with applications in many domains. For example, in data mining, clustering is frequently used to manage, classify, and summarize many kinds of data.

Two widely studied clustering variants are (i) *k-median clustering*, where we compute a set of k centers and the clustering cost is the sum of distances from the data points to their nearest centers, and (ii) *k-means clustering*, where the cost is the sum of squared distances. The *k-median problem* (resp. *k-means problem*) requires computing a set of centers of size k such that the k-median (resp. k-means) clustering cost is minimized.

There have been extensive research on the k-median since 1990s. The k-median problem is shown to be NP-hard in metric spaces by a reduction from dominating set [LV92]. In the metric space, several constant factor approximation algorithms have been proposed for the k-median problem [CGTS02, JV01, AGK$^+$04]. Other interesting variants of these problems are also studied [COP03, Tho05]. In recent years, there has been considerable interest in developing clustering algorithms that can be applied to massive data sets [Ind99, GMMO00, MP04, HM04a, FS05].

For the k-means clustering, one is usually interested in the Euclidean setting. This is mainly because the k-means method [Llo82] works only in Euclidean spaces. See [HM04a] and references therein for further information.

In the first part of the thesis, we present two results for the k-median and k-means clustering. Our results improve the state-of-the-art for the problems considered. In the following, we review the problems studied.

1.1 Coresets for k-median and k-means clustering

Roughly speaking, a *coreset* is a sketch of the data. In the context of clustering, a coreset is a small (weighted) subset from the input, such that for any set C of clustering centers, the cost of clustering the coreset by C is close to the true cost (that is, the cost of clustering the original input set by C).

Computing a (small) coreset is of interest when dealing with massive data sets since, in this situation, applying clustering algorithms on a coreset (rather than the original data set) is more efficient. In particular, for k-median clustering and k-means clustering in $\mathbb{R}^d$, Har-Peled and Mazumdar [HM04a] constructed a coreset of size $O(k\varepsilon^{-d}\log n)$. They used the coreset to compute an $(1+\varepsilon)$ approximation for k-median clustering in $O(n + \rho k^{O(1)} \log^{O(1)} n)$ time, where $\rho = \exp(O([\log(1/\varepsilon)/\varepsilon]^{d-1}))$. Har-Peled and Kushal [HK05] showed that one can construct coresets for this problem with size independent of n.

Streaming. There have been substantial recent interest in performing clustering in the streaming model of computation [GMMO00, COP03, Ind04, HM04a, FS05]. Here points arrive one by one in a stream and one is interested in maintaining a clustering of the points seen so far. Typically, the input is too large to fit in memory. Therefore, it is necessary to maintain a data structure to sketch the data seen so far. In this model, the complexity measure includes the overall space used and the time required to update the data structure. Guha *et al.* [GMMO00] presented an algorithm that uses $O(n^{\varepsilon})$ space to compute a $2^{O(1/\varepsilon)}$-approximation for k-median clustering of points taken from a metric space. Charikar *et al.* [COP03] improved the result significantly by proposing a constant factor approximation algorithm using $O(k \log^2 n)$ space. In the Euclidean settings, Har-Peled and Mazumdar use coresets to compute an $(1+\varepsilon)$-approximation for k-median using $O(k\varepsilon^{-d} \log^{2d+2} n)$ space. The above algorithms handle streams with insertions only. Indyk [Ind04] showed how to handle both insertions and deletions, under the restriction that the points are from a finite resolution grid. Frahling and Sohler [FS05] showed how to extract the coreset quickly and cluster it in the insertion-deletion streaming model, by extending the work of Indyk [Ind04].

1.1.1 Our results.

In Chapter 3, we present fast approximation algorithms for k-clustering using coresets. (For the sake of simplicity of exposition, the phrase *k-clustering* will refer to either k-median or k-means clustering in the remainder of the chapter.) We use a bi-criteria approximation for k-clustering to guide random sampling from the original input. The sampling allows us to extract a (k, ε)-coreset (see Section 2.1 for formal definition) of size (roughly) $O(k^2\varepsilon^{-2} \log^2 n)$ in a general metric space, and a (k, ε)-coreset of size (roughly) $O(k^2 d\varepsilon^{-2} \log n \log(k/\varepsilon))$ in $\mathbb{R}^d$.

In $\mathbb{R}^d$, the small coreset construction leads to an algorithm to find a $(1+\varepsilon)$-approximation to the optimal k-clustering, in $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^{\sigma})$ time (with constant probability of success), for any $\sigma > 0$. This result improves over the algorithm of Kumar *et al.* [KSS04, KSS05], which has running time $O(2^{(k/\varepsilon)^{O(1)}} dn)$. In the streaming model, our main result implies an algorithm that uses $O(dk^2\varepsilon^{-2} \log^8 n)$ space, for $(1+\varepsilon)$-approximation to the optimal k-clustering. The algorithm assumes that the points arrive one by one, and removal of points is not allowed. Upon the arrival of a new point, the amortized time to update the data structure is $O(dk \operatorname{polylog}(ndk/\varepsilon))$. In comparison, previous algorithms require space and time exponential in the dimension.

In a general metric space, the coreset construction leads to a $(10+\varepsilon)$-approximation algorithm for the k-median problem running in $O(nk + k^7\varepsilon^{-4} \log^5 n)$ time using known techniques [AGK⁺01]. This result provides better trade-offs between overall running time and approximation quality over previous results when k is small. In particular, all previous algorithms with $O(nk \operatorname{polylog}(nk))$ running time [Ind99, GMMO00, MP04] provide constant approximation, where the constant is considerably larger than the one in our algorithm. The coreset can also be used to stream k-median clustering using small space, such that one can compute $(1+\varepsilon)$-approximation to the optimal k-median clustering using this data structure. To our knowledge, this is the first algorithm, for general metric spaces, that uses small space and can provide a $(1+\varepsilon)$-approximation to the optimal clustering cost. Of course, since it is not known how to compute the $(1+\varepsilon)$-approximation efficiently (i.e., in polynomial time in n and k), this may be of limited interest.

The main tool we use is random sampling. Mishra *et al.* [MOP01] used a similar approach to obtain a fast k-median algorithm. Their algorithm requires $O((M/\Delta)^2 \log n)$ samples to approximately represent the original input, where n is the input size, M is the diameter of the input, and Δ is the difference between the average clustering cost on the samples and the average clustering cost on the original input. Depending on the parameter M, their algorithm may yield running time as high as $\Omega(n^2)$. Our approach can be interpreted as combining the approach of Mishra *et al.* with the use of coresets and exponential grids of Har-Peled and Mazumdar [HM04a], such that we can obtain "good" samples with size independent of M and with low dependency on the dimension.

These results appeared in [Che06b].

1.2 Approximate k-median clustering with m outliers

Given parameters k and m, we wish to remove a set of at most m points (called *outliers*) from the data set, such that the cost of the optimal k-median clustering of the remaining data is minimized. This is the *k-median with m outliers problem.*

The problem was considered by Charikar *et al.* [CKMN01], and they presented a bi-criteria approximation algorithm for this problem. In particular, their algorithm computes a solution with at most $(1+\lambda)m$ outliers that costs at most $4(1+1/\lambda)OPT$, where OPT is the cost of the optimal solution and $\lambda > 0$ is an arbitrary parameter specified in advance.

This problem arises naturally in situations where noise and errors contained in the data may exert a strong influence over the optimal clustering cost. By removing outliers, one can dramatically reduce the clustering cost and improve the quality of the clustering. In some circumstances, the discovered outliers do not fit the rest of the data, and they are worthy of further investigation. In particular, once identified, they can be used to discover anomalies in the data [RRPS04].

Besides the practical considerations mentioned above, the problem is theoretically interesting. Since the first constant factor approximation algorithm for the k-median problem in metric spaces [CGTS02], there have been numerous developments on this problem and its variants. However, it remained elusive how to design constant factor approximation algorithms for k-median variants that have more than one global constraint. (Indeed, sometimes adding one more global constraint to an optimization problem makes it considerably harder than the original problem. For example, computing the minimum spanning tree is easy, while finding efficient approximation algorithms for k-MST took decades, and the status of bounded degree MST problem has yet to be completely settled [Goe06, SL07].) In particular, as a notable example of such problems, the k-median with outliers problem (MO) has two global constraints imposed by k and m. The MO problem has received considerable interest recently, and coming up with a constant-factor approximation algorithm is a well known open problem, see the discussions in [JV01, CKMN01, Khu05].

Related work. We focus on the most related work here, for further information see [CR05] and references therein.

The *facility location with outliers problem* (FLO for short) is the Lagrangian relaxation of the k-median with outliers problem (MO). Several algorithms [CKMN01, JMM$^+$03, Mah04] were developed for FLO.

Other variants on clustering with outliers include the work of Aboud and Rabani [AR06] that provides an approximation algorithm for a variant of correlation clustering with outliers.

The (uniform) capacitated k-median problem is a k-median variant which have two global constraints. Here we are allowed to open k medians but there is an upper bound on the number of data points each median can serve. There are several bi-criteria approximation algorithms for this problem [CGTS02, BCR01, CR05].

Local search is a popular technique for solving combinatorial optimization problems in practice. Despite their conceptual simplicity, local search algorithms tend to be hard to analyse. It is successfully applied in various facility location problems [KPR00, CG99, AGK$^+$04, ST06] and to the k-median problem [AGK$^+$04]. For some other approximation algorithms that use local search, see [AH98, KR00, KBP03].

1.2.1 Our results

In Chapter 4, we present the first efficient constant factor approximation algorithm for the k-median with outliers problem (MO). Our algorithm is built upon the Lagrangian relaxation framework outlined in [JV01]. It first computes two solutions C_- and C_+ for the *facility location with outliers problem* (FLO), which is the Lagrangian relaxation of MO. Here, C_- has at most k centers and C_+ has at least $k+1$ centers.

In Section 4.2.1, we combine C_- and C_+ into the required approximate solution, when C_+ uses at least $k+2$ facilities. The challenge is to merge a solution with few centers (C_-) which might be too expensive and a solution (C_+) that has too many facilities but is relatively cheap. To confound the difficulty in this "merging" stage, the outliers in these two solutions are not necessarily the same. To perform this "merge", we employ a different greedy algorithm, rather than using the augmentation approaches used in previous approximation algorithms for the k-median problem [JV01, CG99].

We use *successive local search*, in Section 4.2.2, to obtain a constant factor approximation algorithm for MO when C_+ uses $k+1$ facilities. In this case, the cost of C_- cannot be bounded directly by the cost of the optimal solution, and as a result, combining C_- and C_+ into a single solution (as done in previous works [JV01, CG99] and in Section 4.2.1) is no longer viable. To circumvent this difficulty, we use a local search algorithm for the *penalty k-median with outliers problem* (PMO for short) as a subroutine, with gradually increased penalty parameters.

To the best of our knowledge, the use of successive local search, in Section 4.2.2, is new and we consider the introduction of this technique and its analysis to be the main technical contribution of this chapter. Interestingly, neither PMO nor MO can be solved by applying the standard local search methods directly (see Section 2.4). Thus, the new technique seems to be required if one wants to use local search paradigm to solve this problem. Those structural difficulties might explain the challenge in solving this problem, and the complexity of the analysis of our algorithm.

This result will appear in [Che08].

Chapter 2

Preliminaries

In this chapter, we present several technical results that will be used later in the part I.

2.1 Definitions

A *multiset* is a generalization of a set, where an element can occur multiple times. Sometimes, a multiset of points can be represented as a weighted set where the point weights stands for the multiplicity of point. We slightly abuse notations and refer to multisets as sets. Given a set X, the notation $|X|$ refers to the number of distinct elements in X, and the notation $|X|_{\mathsf{w}}$ refers to the total cardinality of X (that is, an element with weight w in X contributes w to $|X|_{\mathsf{w}}$). For an unweighted set, it holds that $|X| = |X|_{\mathsf{w}}$.

We are given a metric space with a distance function $\mathsf{d}(\cdot,\cdot)$ defined over it. We make the standard assumption that we can compute $\mathsf{d}(q,p)$, for any q and p, in constant time. Given a set P of points, a *clustering* of P is a partition induced by a *center set* (or, *facility set*) $C = \{c_1, \ldots, c_k\}$; that is, each point of P is assigned to its nearest neighbor in C. The point $p \in P$ is *served by* c_i if the nearest neighbor to p in C is c_i.

Definition 2.1.1 (k-median and k-means clustering.) The cost of the *k-median clustering* of P by C is

$$\nu(C,P) = \sum_{p \in P} \mathsf{d}(C,p),$$

where $\mathsf{d}(C,p) = \min_{q \in C} \mathsf{d}(q,p)$. The cost of *$k$-means clustering* of P by C is

$$\mu(C,P) = \sum_{p \in P} \Big(\mathsf{d}(C,p)\Big)^2 .$$

Given a set P of points, the metric *k-median* (resp. *k-means*) problem is to find a set of k centers $C \subseteq P$ that minimizes the cost $\nu(C,P)$ (resp. $\mu(C,P)$). Let $\mathrm{opt}_\nu(k,P)$ (resp. $\mathrm{opt}_\mu(k,P)$) denote the cost of the optimal k-median (resp. k-means) clustering of P.

Definition 2.1.2 ((k,ε)-coreset.) Given a set P of points in a metric space, a set $\mathcal{S}$ is a (k,ε)-*coreset* of P for the k-median clustering, if

$$|\nu(C,\mathcal{S}) - \nu(C,P)| \le \varepsilon \nu(C,P),$$

for all sets $C \subseteq P$ satisfying $|C| \le k$. The (k,ε)-*coreset* of P for the k-means clustering is defined similarly.

Definition 2.1.3 ($[\alpha, \beta]$-bi-criteria approximation.) A set $\mathcal{A} = \{a_1, \ldots, a_m\}$ is the center set of an $[\alpha, \beta]$-*bi-criteria approximation* for the k-median (resp. k-means) clustering of P if $m \leq \alpha k$ and $\nu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\nu(k, P)$ (resp. $\mu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\mu(k, P)$).

Given a set P of n points and a set C of facilities, let $\mathbf{N}_{n-m}(C, P)$ be the set of $n - m$ points in P nearest to C. Let

$$\mathsf{A}_m(C, P) = \nu\Big(C, \mathbf{N}_{n-m}(C, P)\Big)$$

be the cost of connecting P to C while excluding the most "expensive" m points from consideration (those m excluded points are the outliers).

Definition 2.1.4 (k-median with m outliers.) Let $\mathsf{MO}(k, P, m)$ be an instance of the k-median with m outliers problem, consisting of an integer $k \geq 1$, a set P of n points, and $m \geq 0$. The objective of $\mathsf{MO}(k, P, m)$ is to compute a set $C \subseteq P$ of k points minimizing the cost $\mathsf{A}_m(C, P)$. Let $\mathrm{opt}_{\mathsf{mo}}(k, P, m)$ denote the cost of the optimal solution.

2.2 Bi-criteria approximation algorithms for metric k-clustering

In this section, we show a bi-criteria approximation algorithm for k-clustering of a point set P in a metric space. The new algorithm is a simple extension of the algorithm of Indyk [Ind99], which computes an $[O(1), O(1)]$-bi-criteria approximation for the k-median problem in $O(nk \,\mathrm{polylog}(nk))$ time, where $n = |P|$. We improve the running time to $O(nk)$ when $k = O(\sqrt{n})$, and show that the same algorithm can also compute a similar approximation for the k-means problem.

The required modifications of Indyk's algorithm are easy and we include the details here only for the sake of completeness.

In the following, we assume that $k = O(\sqrt{n}\,)$ and P is unweighted, see Remark 2.2.4 below. (In fact, if $k = \Omega(\sqrt{n})$, the coreset computed by our algorithm is of size $\Omega(n)$, which is not an interesting case for our coreset construction.)

Let $D(p, q) = \mathsf{d}(p, q)$ when considering the k-median clustering case, and $D(p, q) = (\mathsf{d}(p, q))^2$ when considering the k-means clustering case. That is, in either case, the cost of the clustering with respect to a center set C is $\tau(C, P) = \sum_{q \in P} D(C, q)$.

Claim 2.2.1 *Given points $q_0, q_1, q_2 \in P$, we have that $D(C, q_0) \leq 3(D(C, q_2) + D(q_2, q_1) + D(q_1, q_0))$, for any $C \subseteq P$.*

Proof: Let c be the nearest point to q_2 in C. Then it suffices to prove that

$$D(c, q_0) \leq 3(D(c, q_2) + D(q_2, q_1) + D(q_1, q_0)),$$

since $D(C, q_0) \leq D(c, q_0)$ and $D(C, q_2) = D(c, q_2)$.

If $D(x, y) = \mathsf{d}(x, y)$ then this holds immediately by the triangle inequality. Otherwise, if $D(x, y) = (\mathsf{d}(x, y))^2$, then observe that $D(c, q_2) + D(q_2, q_1) + D(q_1, q_0)$ is minimized when $D(c, q_2) = D(q_2, q_1) =$

FastCluster(k, P)

(i) Sample a set $\mathcal{X}$ of $\mathsf{m} = \left\lceil \mathsf{b}\sqrt{kn\ln k} \right\rceil$ points from P with replacement.

(ii) $C' \leftarrow$ ApproxSlow$(k, \mathcal{X})$.

(iii) Let $\mathcal{Y}$ be the set of m points furthest away from C' in P.

(iv) $C'' \leftarrow$ ApproxSlow$(k, \mathcal{Y})$.

(v) Return $C' \cup C''$.

Figure 2.1: The bi-criteria approximation algorithm for k-clustering of Indyk [Ind99]. Here b is a sufficiently large constant.

$D(q_1, q_0)$ and $\mathsf{d}(c, q_2) + \mathsf{d}(q_2, q_1) + \mathsf{d}(q_1, q_0) = \mathsf{d}(c, q_0)$. Therefore,

$$D(c, q_0) = (\mathsf{d}(c, q_0))^2 = 9\left(\frac{\mathsf{d}(c, q_0)}{3}\right)^2 \leq 3(D(c, q_2) + D(q_2, q_1) + D(q_1, q_0)). \qquad \blacksquare$$

The algorithm FastCluster of Indyk [Ind99] is depicted in Figure 2.1. It requires a "slow" black-box $[\alpha, \beta]$-bi-criteria approximation algorithm ApproxSlow for k-clustering to work. Several known algorithms [JV99, MP04] can serve for this purpose.

Let $\mathrm{opt}_\tau(k, P)$ be the cost of the optimal k-clustering, and let $C_{\mathrm{opt}} = \{c_1, \ldots, c_k\}$ be the set of centers realizing this optimal k-clustering of P. Let K_i denote the cluster of c_i in P, namely, $p \in \mathrm{K}_i$ if c_i is the nearest center to p in C_{opt}. Let $\mathrm{K}'_i = \mathcal{X} \cap \mathrm{K}_i$, where $\mathcal{X}$ is the random sample computed in step (i) of FastCluster. Let $H = \{i \mid |\mathrm{K}_i| \geq \mathsf{m}/k\}$ be the set of indices of the "heavy" clusters. Let $\widehat{\mathrm{K}} = \cup_{i \in H} \mathrm{K}_i$ and $\widehat{\mathrm{K}}' = \cup_{i \in H} \mathrm{K}'_i$. Note that $\left|\widehat{\mathrm{K}}\right| > n - \mathsf{m}$ (indeed, each cluster that is not heavy contains less than m/k points, and there are at most k such clusters).

Claim 2.2.2 ([Ind99]) *Let $\mathcal{E}_1$ be the event that $\tau(C_{\mathrm{opt}}, \mathcal{X}) \leq (1+\varrho)\frac{\mathsf{m}}{n}\tau(C_{\mathrm{opt}}, P)$. We have that $\psi_1 = \mathbf{Pr}[\mathcal{E}_1] \geq \varrho/(1+\varrho)$.*

Proof: Consider an arbitrary sample $q \in \mathcal{X}$. The expected value of $D(C_{\mathrm{opt}}, q)$ is $\tau(C_{\mathrm{opt}}, P)/n$. It follows that $\mathbf{E}[\tau(C_{\mathrm{opt}}, \mathcal{X})] = \frac{|\mathcal{X}|}{n}\tau(C_{\mathrm{opt}}, P) = \frac{\mathsf{m}}{n}\tau(C_{\mathrm{opt}}, P)$. The claim now follows from Markov inequality. $\blacksquare$

Claim 2.2.3 ([Ind99]) *Let $0 < \gamma < 1$ be an arbitrary parameter, and let $\mathcal{E}_2$ be the event that $\frac{|\mathrm{K}_i|}{|\mathrm{K}'_i|} \leq (1+\gamma)\frac{n}{\mathsf{m}}$ for all $i \in H$. We have that $\psi_2 = \mathbf{Pr}[\mathcal{E}_2] \geq 1 - k\exp\left(-\mathsf{m}^2\gamma^2/(8nk)\right)$.*

Proof: Fix an index $i \in H$. It suffices to prove that $\mathbf{Pr}\left[\frac{|\mathrm{K}_i|}{|\mathrm{K}'_i|} > (1+\gamma)\frac{n}{\mathsf{m}}\right] \leq \exp\left(-\frac{\mathsf{m}^2\gamma^2}{8nk}\right)$. Since $1 + \gamma > \frac{1}{1-\gamma/2}$, we have

$$\mathbf{Pr}\left[\frac{|\mathrm{K}_i|}{|\mathrm{K}'_i|} > (1+\gamma)\frac{n}{\mathsf{m}}\right] < \mathbf{Pr}\left[\frac{|\mathrm{K}_i|}{|\mathrm{K}'_i|} > \frac{1}{1-\gamma/2}\frac{n}{\mathsf{m}}\right] = \mathbf{Pr}\left[\frac{|\mathrm{K}'_i|}{|\mathrm{K}_i|} < \left(1 - \frac{\gamma}{2}\right)\frac{\mathsf{m}}{n}\right].$$

Therefore, it suffices to prove that

$$\mathbf{Pr}\left[|\mathrm{K}'_i| \leq \left(1 - \frac{\gamma}{2}\right)\frac{\mathsf{m}}{n}|\mathrm{K}_i|\right] \leq \exp\left(-\frac{\mathsf{m}^2\gamma^2}{8nk}\right),$$

which follows by the Chernoff's inequality, since $|\mathrm{K}_i| \geq \mathsf{m}/k$ (by the definition of H) and $\mathbf{E}[|\mathrm{K}'_i|] = m\,|\mathrm{K}_i|/n$. ■

Consider a function $f_i : \mathrm{K}_i \to \mathrm{K}'_i$, for $i \in H$, such that every point of K'_i has at most $\lceil |\mathrm{K}_i| \,/\, |\mathrm{K}'_i| \rceil$ points assigned to it by f_i. For any point $p \in \mathrm{K}_i$, we have that $D(C', p) \leq 3(D(C', f_i(p)) + D(f_i(p), c_i) + D(c_i, p))$, by Claim 2.2.1. Recall that $\widehat{\mathrm{K}} = \cup_{i \in H} \mathrm{K}_i$ and $\widehat{\mathrm{K}}' = \cup_{i \in H} \mathrm{K}'_i$, and observe that

$$
\begin{aligned}
\tau(C', \widehat{\mathrm{K}}) &= \sum_{i \in H} \sum_{p \in \mathrm{K}_i} D(C', p) \leq 3 \sum_{i \in H} \sum_{p \in \mathrm{K}_i} \Big[D(C', f_i(p)) + D(f_i(p), c_i) + D(c_i, p) \Big] \\
&\leq 3 \sum_{i \in H} \left\lceil \frac{|\mathrm{K}_i|}{|\mathrm{K}'_i|} \right\rceil \sum_{q \in \mathrm{K}'_i} \Big[D(C', q) + D(q, c_i) \Big] + 3\tau(C_{\mathrm{opt}}, \widehat{\mathrm{K}}) \\
&\leq 3(1+\gamma) \frac{n}{\mathsf{m}} \sum_{i \in H} \sum_{q \in \mathrm{K}'_i} \Big[D(C', q) + D(C_{\mathrm{opt}}, q) \Big] + 3\tau(C_{\mathrm{opt}}, \widehat{\mathrm{K}}) \\
&= 3 \Big[(1+\gamma) \frac{n}{\mathsf{m}} \Big(\tau(C', \widehat{\mathrm{K}}') + \tau(C_{\mathrm{opt}}, \widehat{\mathrm{K}}') \Big) + \tau(C_{\mathrm{opt}}, \widehat{\mathrm{K}}) \Big] \\
&\leq 3 \Big[(1+\gamma) \frac{n}{\mathsf{m}} \Big(\tau(C', \mathcal{X}) + \tau(C_{\mathrm{opt}}, \mathcal{X}) \Big) + \tau(C_{\mathrm{opt}}, P) \Big],
\end{aligned}
$$

where the second inequality holds because for every $q \in \mathrm{K}'_i$ there are at most $\lceil |\mathrm{K}_i| \,/\, |\mathrm{K}'_i| \rceil$ points of K_i assigned to it by f_i, the third inequality holds with probability ψ_2 by Claim 2.2.3, and the last inequality holds because $\widehat{\mathrm{K}} \subseteq P$ and $\widehat{\mathrm{K}}' \subseteq \mathcal{X}$. Since ApproxSlow is an $[\alpha, \beta]$-bi-criteria approximation algorithm for k-clustering, we have $\tau(C', \mathcal{X}) \leq \beta\, \mathrm{opt}_\tau(k, \mathcal{X}) \leq \beta \tau(C_{\mathrm{opt}}, \mathcal{X})$. It thus follows that

$$
\begin{aligned}
\tau(C', \widehat{\mathrm{K}}) &\leq 3 \Big[(1+\gamma) \frac{n}{\mathsf{m}} \Big(\tau(C', \mathcal{X}) + \tau(C_{\mathrm{opt}}, \mathcal{X}) \Big) + \tau(C_{\mathrm{opt}}, P) \Big] \\
&\leq 3 \Big[(1+\gamma) \frac{n}{\mathsf{m}} (1+\beta) \tau(C_{\mathrm{opt}}, \mathcal{X}) + \tau(C_{\mathrm{opt}}, P) \Big] \\
&\leq 3 \Big[(1+\gamma) \frac{n}{\mathsf{m}} (1+\beta)(1+\varrho) \frac{\mathsf{m}}{n} \tau(C_{\mathrm{opt}}, P) + \tau(C_{\mathrm{opt}}, P) \Big] \\
&= 3((1+\gamma)(1+\beta)(1+\varrho) + 1)\, \mathrm{opt}_\tau(k, P),
\end{aligned} \tag{2.1}
$$

where the last inequality holds with probability ψ_1, by Claim 2.2.2.

Because $\left| \widehat{\mathrm{K}} \right| \geq n - \mathsf{m}$, the cost of points in $P \setminus \mathcal{Y}$ with respect to center set C' does not exceed $\tau(C', \widehat{\mathrm{K}})$. Indeed, the points of $\mathcal{Y}$ are the m most expensive points in P with respect to C', and as such we have

$$
\tau(C', P \setminus \widehat{\mathrm{K}}) = \sum_{p \in P \setminus \widehat{K}} D(C', p) \leq \sum_{p \in \mathcal{Y}} D(C', p) = \tau(C', \mathcal{Y}),
$$

since $\left| P \setminus \widehat{\mathrm{K}} \right| \leq \mathsf{m}$. This implies

$$
\tau(C', P \setminus \mathcal{Y}) = \tau(C', P) - \tau(C', \mathcal{Y}) \leq \tau(C', P) - \tau(C', P \setminus \widehat{\mathrm{K}}) = \tau(C', \widehat{K}).
$$

In addition, we have that $\tau(C'', \mathcal{Y}) \leq \beta\, \mathrm{opt}_\tau(k, \mathcal{Y}) \leq \beta\, \mathrm{opt}_\tau(k, P)$. Therefore, by Eq. (2.1),

$$
\begin{aligned}
\tau(C' \cup C'', P) &\leq \tau(C', P \setminus \mathcal{Y}) + \tau(C'', \mathcal{Y}) \leq \tau(C', \widehat{K}) + \beta\, \mathrm{opt}_\tau(k, P) \\
&\leq (3((1+\gamma)(1+\beta)(1+\varrho) + 1) + \beta)\, \mathrm{opt}_\tau(k, P) \\
&< 3(1+\gamma)(1+\beta)(2+\varrho)\, \mathrm{opt}_\tau(k, P).
\end{aligned}
$$

Set $\gamma = 1/5$ and $\varrho = 3$. We have that $\psi_1 \geq \varrho/(1+\varrho) = 3/4$ and

$$\psi_2 \geq 1 - k \exp\left(-\frac{\mathsf{m}^2\gamma^2}{8nk}\right) \geq 1 - k \exp\left(-\frac{(\mathsf{b}\sqrt{kn\ln k}\,)^2\gamma^2}{8nk}\right) = 1 - k \exp\left(-\frac{\mathsf{b}^2 \ln k}{128}\right) \geq \frac{3}{4},$$

for $\mathsf{b} \geq 20$, by Claim 2.2.2 and Claim 2.2.3. It follows that the algorithm succeeds with probability $\mathbf{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2] \geq \psi_1 + \psi_2 - 1 \geq 1/2$. (Note that $\mathcal{E}_1$ and $\mathcal{E}_2$ are not necessarily independent.)

Since $|C' \cup C''| \leq 2\alpha k$ and $3(1+\gamma)(1+\beta)(2+\varrho) = 24\beta+24$, It follows that the algorithm FastCluster computes a $[2\alpha, 24\beta + 24]$-bi-criteria approximation for k-clustering, with constant probability. If the black-box algorithm ApproxSlow runs in $g(n)$ time, then the new algorithm runs in time $O(nk + g(\sqrt{kn\ln k}\,))$. Note that we can boost the probability of success to be arbitrarily close to 1 by increasing ϱ and b (this of course would result in a worse approximation).

The result. The algorithm of Jain and Vazirani [JV99] can be used as the black-box algorithm ApproxSlow inside FastCluster, and we get a new algorithm, denoted by IJVAlg, which runs in time $O(nk \log k \log^2 n)$. (Note that the algorithm of Jain and Vazirani works for both k-median and k-means clustering.) Now, use IJVAlg as the black-box algorithm in FastCluster. The resulting algorithm returns a $[O(1), O(1)]$-bi-criteria approximation, and the overall running time is $O\big(nk + \sqrt{kn\log k} \cdot k \log k \log^2 n\big) = O(nk)$, since by assumption $k = O(\sqrt{n})$. Since the algorithm IJVAlg might fail, we boost its success probability to, say, above 0.99 (by increasing ϱ and b as suggested above). It is now easy to verify that the new algorithm succeeds with probability $\geq 1/2$.

Remark 2.2.4 If P is weighted, with total weight W, we use the grouping technique of Mettu and Plaxton [MP04]. We group points with roughly equal weights together, run the unweighted algorithm on each group with confidence parameter set to $O(1/\log W)$, and combine the centers computed for each group. See [MP04] for details. This yields a constant factor approximation using $O(k \log W)$ centers with constant probability. The overall running time is $O(nk \log\log W)$.

Note that for the above algorithm, we can boost its success probability from constant to $\geq 1 - \lambda/2$ by running it $O(\log(1/\lambda))$ times, and take the best solution computed (that is, the solution with the cheapest cost). We summarize.

Theorem 2.2.5 *Given a set P of n points in a metric space and a parameter $k = O(\sqrt{n})$, one can compute $O(k)$ centers in $O(nk\log(1/\lambda))$ time, such that the cost of k-median clustering of P using these centers is within a constant factor of the optimal k-median clustering cost. The algorithm succeeds with probability $\geq 1 - \lambda/2$.*

If the input is weighted, with total weight W, the algorithm computes $O(k \log W)$ centers, and the running time is $O(nk \log(1/\lambda) \log\log W)$.

The same result holds verbatim for k-means clustering.

2.3 An approximation algorithm for facility location with m outliers

In this section, we present an approximation algorithm FLOALG for the facility location with m outliers problem, which is the Lagrangian relaxation of the k-median with m outliers (MO). The algorithm presented here is due to Charikar *et al.* [CKMN01].

Definition 2.3.1 (Facility location with m outliers.) Let $\mathsf{FLO}(z, P, m)$ be an instance of facility location with m outliers, consisting of a parameter $z \geq 0$, a set P of points, and an integer $m \geq 0$. The objective of $\mathsf{FLO}(z, P, m)$ is to compute a set $C \subseteq P$ minimizing the cost $\mathsf{A}_m(C, P) + z\,|C|$. Let $\mathrm{opt}_{\mathsf{flo}}(z, P, m)$ denote the cost of the optimal solution.

The input to the algorithm is a set P of n points, a set F of facilities (we assume that $F = P$ here), the cost z for opening a facility, and the parameter m. For convenience, we assume that $P = \{1, \ldots, n\}$.

There is a natural integer program (IP) for this problem. The IP has a variable $y_f \in \{0, 1\}$ indicating if a facility $f \in F$ is opened or not, a variable $x_{vf} \in \{0, 1\}$ indicating if a point $v \in P$ is served by f, and $o_v \in \{0, 1\}$ indicating if $v \in P$ is considered to be an outlier. We require that if v is served by f (i.e., $x_{vf} = 1$) then the facility f is opened (i.e., $y_f = 1$). In addition, any client is either served by some facility, or it is considered to be an outlier. Finally, the total number of outliers cannot exceed m. Given a feasible solution to this integer program, its cost is $\sum_{v,f} \mathsf{d}(v, f) \cdot x_{vf} + z \sum_f y_f$. In particular, the optimal solution for the IP with this objective function is the optimal solution for FLO.

The LP relaxation of this IP is

$$
\begin{array}{rrcl}
 & \min & & \displaystyle\sum_{v,f} \mathsf{d}(v,f) \cdot x_{vf} + z \sum_f y_f \\
\forall v \in P, f \in F, & x_{vf} & \leq & y_f \\
\forall v \in P, & 1 & \leq & \displaystyle\sum_f x_{vf} + o_v \\
 & \displaystyle\sum_v o_v & \leq & m \\
 & x_{vf}, y_f, o_v & \geq & 0.
\end{array}
$$

Rewriting it in general form, we have

$$
\begin{array}{rrcl}
 & \min & & \displaystyle\sum_{v,f} \mathsf{d}(v,f) \cdot x_{vf} + z \sum_f y_f \\
\forall v \in P, f \in F, & y_f - x_{vf} & \geq & 0 \\
\forall v \in P, & \displaystyle\sum_f x_{vf} + o_v & \geq & 1 \\
 & -\displaystyle\sum_v o_v & \geq & -m \\
 & x_{vf}, y_f, o_v & \geq & 0.
\end{array}
$$

Now, the dual of the above linear program is:

$$
\begin{array}{rrcll}
 & \max & & \sum_v \alpha_v - m\gamma & (2.2)\\
\forall v \in P, f \in F, & \alpha_v - \beta_{vf} & \leq & \mathsf{d}(v,f) & \text{(i)}\\
\forall f \in F, & \sum_v \beta_{vf} & \leq & z & \text{(ii)}\\
\forall v \in P, & \alpha_v - \gamma & \leq & 0 & \text{(iii)}\\
 & \alpha_v, \beta_{vf}, \gamma & \geq & 0. &
\end{array}
$$

The algorithm. The algorithm FloAlg [CKMN01] is based on the constant-factor approximation algorithm for k-median by Jain and Vazirani [JV01]. It works in two stages.

- **First stage.** Initially all clients are labeled as outliers, and $\alpha_v = 0$, for all $v \in P$. The dual variables $\alpha_1, \ldots, \alpha_n$ grow uniformly. In particular, at time $t > 0$, if α_v is still growing then $\alpha_v = t$. When $\alpha_v > \mathsf{d}(v,f)$, the edge vf is *saturated*. Set

$$
\beta_{vf} = \begin{cases} 0 & \text{if } \alpha_v \leq \mathsf{d}(v,f) \\ \alpha_v - \mathsf{d}(v,f) & \text{otherwise.} \end{cases}
$$

  As this process continues, the following events may be encountered:

  (A) If $\sum_v \beta_{vf} = z$, for some $f \in F$, then the facility f is *paid for*. In this case, for all $v \in P$ with $\beta_{vf} > 0$, the point v ceases to be an outlier, α_v stops growing, and as such, f is designated as the *witness* of v (provided that v has not been assigned a witness already). Let t_f be the time at which the facility f gets paid for.

  (B) If $\alpha_v = \mathsf{d}(v,f)$ and f is already paid for, for some $v \in P$ and $f \in F$, then v ceases to be an outlier, α_v stops growing and f is the *witness* of v.

  This stage terminates when the number of outliers drops below m for the first time (note that all points are initially labeled as outliers, and when an event occurs, at least one point stop being labeled as an outlier). In the last step, either an event (A) or an event (B) occurs. If an event (B) occurs, then we have exactly m outliers. If an event (A) occurs, however, some facility f' gets paid for and a set $N_{f'}$ of clients are assigned f' as their witness, and the number of outliers may be strictly less than m. To obtain exactly m outliers, we select an arbitrary subset of points from $N_{f'}$ to be reassigned as outliers such that the total number of outliers is exactly m.

  Let U be denote the set of m outliers, and let X denote the set of facilities which get paid for in the end of this stage.

- **Second stage.** The algorithm decides which facilities get opened among X (intuitively, X may contain facilities that are "close" to each other, and we need only one of them for the solution). Specifically, we form a graph G by connecting a facility $f \in X$ with a client $v \in P$ if vf is saturated (that is, $\alpha_v > \mathsf{d}(v,f)$). Next, we repeatedly pick the facility $f \in X$ that is the earliest paid for (among these that were not selected yet), and remove all facilities in X that have a common client connected to both of them (in the graph G). In the end of this process, every client is connected to a single facility.

Let $C \subseteq X$ denote the set of facilities that survive this filtering process. The set C is the solution returned by FloAlg. (The set of outliers in the solution for C is the set U computed in the first stage.)

2.3.1 Correctness

Let $N = P \setminus U$ be the set of all clients excluding the m outliers. Note that f' (the last facility that got paid for in the first stage of FloAlg) may or may not be in C.

Lemma 2.3.2 *We have that* $\sum_{v \in N} \alpha_v \leq \mathrm{opt}_{\mathsf{flo}}(z, P, m)$*, where* $\mathrm{opt}_{\mathsf{flo}}(z, P, m)$ *denotes the cost of the optimal solution for* $\mathsf{FLO}(z, P, m)$.

Proof: Let t' be the time at which the first stage terminates (this is the time when at most m clients are labeled as outliers). At this point, FloAlg maintains values for all the variables of the dual LP (except for γ). By setting the dual variable $\gamma = t'$, we argue that this results in a feasible solution to the dual LP above. Note that $\alpha_v \leq t'$, for all $v \in P$, satisfying the constraint (iii) in Eq. (2.2). The constraints (i) and (ii) in Eq. (2.2) are also satisfied, since these inequalities are always maintained in the first stage of FloAlg. As such, this is a feasible solution for the dual LP. The value of this dual solution is

$$\sum_{v \in P} \alpha_v - m\gamma = \sum_{v \in P} \alpha_v - \sum_{v \in U} \alpha_v = \sum_{v \in N} \alpha_v,$$

because $|U| = m$ and $\alpha_v = t' = \gamma$, for every point $v \in U$. Denote by OPT_D the value of the optimal solution for the dual LP, and denote by OPT_P the value of the optimal solution for the primal LP, we have $\sum_v \alpha_v - m\gamma \leq OPT_D \leq OPT_P$, by the weak duality theorem. Now, since the optimal solution for $\mathsf{FLO}(z, P, m)$ is a feasible (integral) solution for the primal LP, we have $OPT_P \leq \mathrm{opt}_{\mathsf{flo}}(z, P, m)$. ■

Definition 2.3.3 A client $v \in N$ is *directly connected* to C if there exists a facility $f \in C$ such that $\beta_{vf} > 0$. In this case v is assigned to f. (Note that v can only be directly connected to at most one facility in C, because the second stage removes such facilities from consideration.)

On the other hand, if there is no facility $f \in C$ such that $\beta_{vf} > 0$, then the client v is either assigned to its witness (which happens during an event (B) in the first stage of FloAlg) or to the facility that caused the deletion of its witness in the second stage. Such clients are *indirectly connected* to C.

Observation 2.3.4 *A facility gets paid for only at an event (A) during the execution of* FloAlg.

Claim 2.3.5 *Let $s \in P$ be a client and $g \in X$ be a facility such that sg is saturated, namely, $\alpha_s > \mathsf{d}(s, g)$. Then, when* FloAlg *terminates, we have that $t_g \geq \alpha_s$, where t_g is the time at which g is paid for.*

Proof: If $t_g \leq \mathsf{d}(s, g)$, then at time $t = \mathsf{d}(s, g)$, the value of α_s is $t = \mathsf{d}(s, g)$, and the facility g is already paid for. As such, FloAlg triggers an event (B) for s, and α_s stops growing immediately, yielding $\alpha_s = \mathsf{d}(s, g)$. This contradicts the assumption that $\alpha_s > \mathsf{d}(s, g)$.

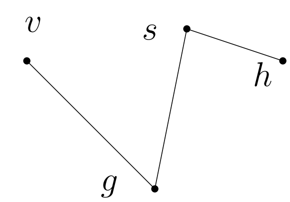
Thus, it holds that $t_g > \mathsf{d}(s, g)$. Assume for the sake of contradiction that $t_g < \alpha_s$. Then, at time $t = t_g$, an event (A) occurs for g, the facility g is designated as the witness of s, and α_s stops growing, yielding $\alpha_s = t_g$. A contradiction. ■

Claim 2.3.6 *If $g \in X$ is the witness for a client $v \in N$, then $\alpha_v \geq \mathsf{d}(v, g)$ and $\alpha_v \geq t_g$, where t_g is the time at which g is paid for.*

Proof: Observe that v can be assigned to its witness g only after α_v exceeds $\mathsf{d}(v, g)$. Similarly, v can have g as its witness only after g got paid for. ■

Claim 2.3.7 *If v is indirectly connected to $h \in C$, then we have that $\mathsf{d}(v, C) \leq 3\alpha_v$.*

Proof: First, consider the case that h is the witness for v. If h is designated as witness for v during an event (A), then $\beta_{vh} > 0$, but this implies that v is directly connected to C. A contradiction. As such, the facility h was designated as the witness for v in an event (B). But then $\alpha_v = \mathsf{d}(v, h)$. It follows that $\mathsf{d}(v, C) \leq \mathsf{d}(v, h) = \alpha_v$.



Otherwise, let $g \neq h$ denote the witness for v. Observe that g must have been removed from g in the second stage of FLOALG. Specifically, this removal is initiated because there is a point, say s, such that edges sg and sh are both saturated.

Let t_g denote the time at which g is paid for. By Claim 2.3.6, we have $\alpha_v \geq t_g$. In addition, by Claim 2.3.5, we have $t_g \geq \alpha_s$. Combining these two inequalities, we obtain

$$\alpha_v \geq t_g \geq \alpha_s. \tag{2.3}$$

Now, since sg and sh are both saturated, we have, by definition, that $\alpha_s > \mathsf{d}(s, g)$ and $\alpha_s > \mathsf{d}(s, h)$. Finally, we have, by Claim 2.3.6, that $\alpha_v \geq \mathsf{d}(v, g)$. Therefore, it follows from Eq. (2.3) and the triangle inequality that

$$3\alpha_v \geq \alpha_v + 2\alpha_s \geq \mathsf{d}(v, g) + \mathsf{d}(s, g) + \mathsf{d}(s, h) \geq \mathsf{d}(v, h) \geq \mathsf{d}(v, C).$$ ■

Observation 2.3.8 *Let $v \in N$ and $h \in C$.*

(i) If v is directly connected to h, then $\beta_{vf} = 0$ for all $f \neq h$ in C.

(ii) If v is indirectly connected to h, then $\beta_{vf} = 0$ for all $f \in C$.

Proof: (i) Note that if $\beta_{vf} > 0$ then f would have been removed from X (due to h) in the second stage of FLOALG. As such, f cannot be in C, as $C \subseteq X$.

(ii) Otherwise, by definition, v is directly connected to some facility in C. ■

Lemma 2.3.9 *We have that $\sum_{v \in N} \mathsf{d}(v, C) + 3z(|C| - 1) \leq 3 \sum_{v \in N} \alpha_v$.*

Proof: For every $f \in C \setminus \{f'\}$, we have that $z = \sum_{v \in N} \beta_{vf}$. Indeed, when $z = \sum_{v \in P} \beta_{vf}$ holds (at which time an event (A) occurs), the algorithm adds to N all the clients with saturated edges to f. In particular, we have $\beta_{vf} = 0$, for $v \in U$. Therefore,

$$z(|C| - 1) \leq \sum_{f \in C \setminus \{f'\}} \sum_{v \in N} \beta_{vf} = \sum_{v \in N} \sum_{f \in C \setminus \{f'\}} \beta_{vf}. \tag{2.4}$$

(Note that if $f' \notin C$ then the inequality in the above equation holds as a strict inequality.) Next, we claim that

$$\sum_{v \in N} \mathsf{d}(v, C) + 3 \sum_{v \in N} \sum_{f \in C \setminus \{f'\}} \beta_{vf} \le 3 \sum_{v \in N} \alpha_v. \tag{2.5}$$

Indeed, consider an arbitrary client $v \in N$, and assume that v is connected to the facility h.

(i) If v is directly connected to h, then $\alpha_v = \mathsf{d}(v, h) + \beta_{vh}$, and moreover, $\beta_{vf} = 0$ for any $f \neq h$ in C, by Observation 2.3.8. Thus, its contribution to the left-handed side of Eq. (2.5) above is $\mathsf{d}(v, C) + 3\beta_{vh} \le 3\mathsf{d}(v, C) + 3\beta_{vh} \le 3\alpha_v$.

(ii) If v is indirectly connected to h, then we have $\beta_{vf} = 0$ for any $f \in C$, by Observation 2.3.8. Therefore, its contribution to the left-handed side of Eq. (2.5) above is $\mathsf{d}(v, C) \le 3\alpha_v$, by Claim 2.3.7.

Now, by Eq. (2.4) and Eq. (2.5), it follows that

$$\sum_{v \in N} \mathsf{d}(v, C) + 3z(|C| - 1) \le \sum_{v \in N} \mathsf{d}(v, C) + 3 \sum_{v \in N} \sum_{f \in C \setminus \{f'\}} \beta_{vf} \le 3 \sum_{v \in N} \alpha_v.$$

■

Running time. The algorithm FloAlg maintains a queue of events, at each iteration handling the event in the queue with the smallest time stamp. Initially, for each client $v \in P$, we store every edge vf in the queue with time stamp $\mathsf{d}(v, f)$ (this corresponds to an event (B) associated with v). For each "active" (i.e., not paid for yet) facility $f \in F$, we maintain the time after which it will get paid for, and this corresponds to the event (A) associated with f. Now, we sort all the events in order of increasing time stamp, and when the earliest event occurs, we update (or, reschedule) its related events. With careful (but straightforward) implementation, the algorithm handles $O(n^2)$ events, and the overall running time is $O(n^2 \log n)$, since a queue operation takes logarithmic time.

The result. Lemma 2.3.2 and Lemma 2.3.9 together with the above discussion establish the following theorem.

Theorem 2.3.10 ([CKMN01]) *Given a set P of n points and $z \ge 0$, one can compute, in $O(n^2 \log n)$ time, a facility set $C \subseteq P$ such that $\mathsf{A}_m(C, P) + 3z(|C| - 1) \le 3\mathrm{opt}_{\mathsf{flo}}(z, P, m)$.*

2.4 A counter example for using standard local search for MO

In this section, we show that using the standard *local search* method for the k-median clustering with m outliers (MO) may yields arbitrarily bad performance. This example is due to Yusu Wang.

The local search method uses the concept of neighborhood. Specifically, it starts with an initial feasible solution and then repeatedly searches the neighborhood of the current solution for a better solution until it cannot be improved any further (that is, it reaches a *locally optimal* solution).

For the k-median clustering problem, Arya *et al.* [AGK+04] analysed a simple local search algorithm KmLocalSearch, described in Figure 2.2, and showed that it provides a constant factor approximation for the k-median clustering.

```
Algorithm KmLocalSearch(P)
   S ← an arbitrary set of k centers
   while ∃ S' ∈ N(S) such that ν(S', P) < ν(S, P) do
       S ← S'
   return S
```

Figure 2.2: A local search algorithm for the k-median clustering problem. Here P is the input point set, and $\mathtt{N}(S) = \{(S \setminus \{s\}) \cup \{s'\} \mid s \in S\}$.

Theorem 2.4.1 ([AGK⁺04]) *The algorithm* KmLocalSearch *computes, in $O(n^2 \log n)$ time, a 5-approximation for the k-median clustering problem.*

The algorithm KmLocalSearch can be easily extended to the k-median clustering with m outliers (MO). In particular, given a solution S, which is a set of k facilities, a *neighbor* X of S is a set of k facilities such that $|X \cap S| \geq k - \mathsf{b}$, for some constant $\mathsf{b} \geq 0$. Namely, X is obtained by swapping at most b facilities of S with facilities outside S. Let $\mathtt{N}(S)$ denote the set of all neighbors of S. As in KmLocalSearch, the local search algorithm repeatedly replace S by a better center set in $\mathtt{N}(S)$ as long as such center set exists.

In the following, we present an example demonstrating that a locally optimal solution yielded by this standard local search algorithm may have arbitrarily bad performance, compared to the globally optimal solution (namely, the *locality gap* can be arbitrarily large).

Suppose that $n \gg m \gg k > 1$, and $u = m/(k-1)$ is an integer. Consider an input P as follows: the set P is partitioned into disjoint subsets B, $C_1, \ldots, C_{k-1}$, $D_1, \ldots, D_{k-2}$, and E, such that the distance between any pair of points belonging to different subsets is very large. Suppose that

(i) $|B| = n - 2m$ and $\mathsf{d}(p,q) = 0$, for any $p, q \in B$.

(ii) For each $i = 1, \ldots, k-1$, we have $|C_i| = u$ and $\mathsf{d}(p,q) = \beta$, for any $p, q \in C_i$.

(iii) For each $j = 1, \ldots, k-2$, we have $|D_j| = u - 1$ and $\mathsf{d}(p,q) = 0$, for any $p, q \in D_j$.

(iv) $|E| = u + k - 2$ and $\mathsf{d}(p,q) = \gamma$, for any $p, q \in E$.

We further assume that $u \gg k$ and $\gamma < (u-1)\beta < 2\gamma$.

For $Y \in \{B, C_1, \ldots, C_{k-1}, D_1, \ldots, D_{k-2}, E\}$, let $f(Y)$ denote an arbitrary point in Y. Consider a solution $\mathcal{S} = \{f(B), f(D_1), \ldots, f(D_{k-2}), f(E)\}$, namely, we place a facility in each of the subsets $B, D_1, \ldots, D_{k-2}, E$. The $m = (k-1)u$ outliers in this solution are the points in $C_1, \ldots, C_{k-1}$.

Claim 2.4.2 *The solution $\mathcal{S}$ is locally optimal, incurring a cost of $(u + k - 3)\gamma$, if $\mathsf{b} < k - 1$.*

Proof: In the solution $\mathcal{S}$, serving B by $f(B)$ costs 0, serving D_j by $f(D_j)$ costs 0, for $j = 1, \ldots, k-2$, and serving E by $f(E)$ costs $(u + k - 3)\gamma$. Therefore, the cost of $\mathcal{S}$ is $(u + k - 3)\gamma$.

To see that $\mathcal{S}$ is locally optimal, observe that we cannot swap $f(B)$ out, because otherwise we need to serve points in B by a facility not in B (recall that $|B| = n - 2m$ and $n \gg m$), which is very costly. For the same reason, we cannot swap $f(E)$ out. Suppose that we swap $\mathsf{b}' \leq \mathsf{b}$ facilities, say $f(D_1), \ldots, f(D_{\mathsf{b}'})$, with $f(C_1), \ldots, f(C_{\mathsf{b}'})$, and let $\mathcal{S}'$ denote the resulting solution. That is,

$$\mathcal{S}' = \{f(B), f(C_1), \ldots, f(C_{\mathsf{b}'}), f(D_{\mathsf{b}'+1}), \ldots, f(D_{k-2}), f(E)\}.$$

It is easy to verify that the cost of $\mathcal{S}'$ is $\mathsf{b}' \cdot (u-1)\beta + (u+k-3-\mathsf{b}')\gamma$, which is greater than $(u+k-3)\gamma$ since $(u-1)\beta > \gamma$. This implies that we cannot improve $\mathcal{S}$ by swapping $\leq \mathsf{b}$ facilities. ■

It is easy to verify that the optimal solution is $\overline{\mathcal{S}} = (B, C_1, \ldots, C_{k-1})$, which has a cost of $(k-1)(u-1)\beta$. Since $u \gg k$ and $(u-1)\beta$ is only slightly larger than γ, it follows that the locality gap

$$\frac{(u+k-3)\gamma}{(k-1)(u-1)\beta} > \frac{u+k-3}{2(k-1)},$$

since $(u-1)\beta < 2\gamma$ (by assumption). This may be arbitrarily large, depending on the ratio u/k.

Now, note that $\mathsf{MO}(k, P, m)$ can be reduced to $\mathsf{PMO}(k, P, \varrho, m)$, by setting $\varrho = \infty$. Since $\mathsf{MO}(k, P, m)$ cannot be solved by the above local search algorithm, neither can $\mathsf{PMO}(k, P, \varrho, m)$.

2.5 Perturbation of the distance function d

In this section, we show that, given a set P of n points, one can slightly perturb the distance function d defined over P (for the purpose of an approximation algorithm for the problems studied), so that the distances between all pairs of points in P are distinct, and the spread of P is polynomially bounded.

Since the k-median clustering problem can be reduced to the k-median clustering with m outliers problem (MO), by setting $m = 0$, we only show the perturbation scheme for MO.

Let opt be the cost of the optimal solution for $\mathsf{MO}(k, P, m)$. We first compute a real number σ as an estimate of opt.

Lemma 2.5.1 *One can compute in polynomial time a real number σ such that $\mathsf{opt}/(3n) \leq \sigma \leq \mathsf{opt}$.*

Proof: The problem of *k-center with m outliers* (CO for short) is to compute a set of m outliers so as to minimize the cost of the *k-center* clustering of the remaining points. Let $\mathrm{opt}_{\mathsf{co}}(P, k, m)$ be the cost of the optimal solution for the CO instance with input point set P. It is easy to verify that $\mathsf{opt}/n \leq \mathrm{opt}_{\mathsf{co}}(P, k, m) \leq \mathsf{opt}$. We use the algorithm for CO presented in [CKMN01] to compute β such that $\beta/3 \leq \mathrm{opt}_{\mathsf{co}}(P, k, m) \leq \beta$. The claim now follows by setting $\sigma = \beta/3$. ■

Given parameters $0 < \varepsilon < 1$ and $1 \leq \gamma$, we shall perturb the distance function d, and denote the resulting new distance function by d'. We claim that if one can compute a set C of k facilities such that $\mathsf{A}_m(\mathsf{C}, P) \leq \gamma\mathsf{opt}$ under the distance function d', then it holds $\mathsf{A}_m(\mathsf{C}, P) \leq (1+\varepsilon)\gamma\mathsf{opt}$ under the original distance function d. We omit the easy proof here, and only specify the perturbation scheme in the following.

Let $\Delta = \varepsilon\sigma/(2n)$, and let $\tau(p, q)$ be a small random real number such that $0 \leq \tau(p, q) \leq \Delta/2$, for all $p, q \in P$ (note that $\tau(p, q)$ is independent for every $p, q \in P$). For each pair $p, q \in P$, if $\mathsf{d}(p, q) > 6\gamma n\sigma$ then let $\mathsf{d}'(p, q) = 6\gamma n\sigma + \Delta + \tau(p, q)$, otherwise, let $\mathsf{d}'(p, q) = \mathsf{d}(p, q) + \Delta + \tau(p, q)$. It is easy to verify that, under the distance function d', the inter-point distances are distinct (with high probability), the ratio between the maximum inter-point distance and the minimum inter-point distance is $O(\gamma n^2/\varepsilon)$, and $\mathsf{d}'(x, y) + \mathsf{d}'(y, z) \geq \mathsf{d}'(x, z)$ holds for every $x, y, z \in P$.

Chapter 3

Coresets for k-median and k-means Clustering

3.1 Introduction

In this chapter, we present fast approximation algorithms for k-clustering using coresets. In Section 3.2, we construct a (k, ε)-coreset of size (roughly) $O(k^2 \varepsilon^{-2} \log^2 n)$ in a general metric space. This is the first coreset construction that works for a general metric space. In Section 3.3, we present an algorithm to extract a (k, ε)-coreset of size (roughly) $O(dk^2 \varepsilon^{-2} \log n \log(k/\varepsilon))$ in $\mathbb{R}^d$. This is the first coreset construction with *polynomial* dependency in the dimension. In Section 3.4, we extend the coreset constructions to the k-means clustering. We provide some applications of those coresets in Section 3.5.

3.2 Coreset for metric k-median clustering

In this section, we present an algorithm to compute a (k, ε)-coreset for metric k-median clustering. The input consists of a set P of n points and parameters k, ε, and λ. There is also an associated metric distance function d defined over the points of P, which we can evaluate for any pair of points of P in constant time. We shall compute a weighted sample set $\mathcal{S}$ from P such that $\mathbf{w}(\mathcal{S}) = \mathbf{w}(P)$ and $\mathcal{S}$ is a (k, ε)-coreset of P, with probability $\geq 1 - \lambda$.

3.2.1 The coreset construction

The algorithm consists of two steps: (i) partitioning the input P into several disjoint subsets, and (ii) taking a random sample from each such subset. The union of these samples form the desired coreset.

Step 1: Partitioning P

For the sake of simplicity of exposition, we assume that the input P is unweighted unless explicitly stated otherwise. The results hold also when P is weighted, with slightly worse bounds.

Assume that $\mathcal{A} \subseteq P$ is the center set of an $[\alpha, \beta]$-bi-criteria approximation to the optimal k-median clustering of P. That is, $\mathcal{A} = \{a_1, \ldots, a_m\}$ satisfies $\nu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\nu(k, P)$, where $m \leq \alpha k$ and $\alpha, \beta \geq 1$ (here, α and β are some constants).

Let $P_i \subseteq P$ be the set of points served by the center a_i, for $i = 1, \ldots, m$, and let $R = \nu(\mathcal{A}, P)/(\beta n)$ be a lower bound on the average radius of the optimal k-median clustering. Set $\phi = \lceil \lg(\beta n) \rceil$, where $\lg x = \log_2 x$. For $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$, let

$$P_{i,j} = \begin{cases} P_i \cap \mathrm{ball}(a_i, R) & j = 0 \\ P_i \cap \left[\mathrm{ball}(a_i, 2^j R) \setminus \mathrm{ball}(a_i, 2^{j-1} R) \right] & j \geq 1 \end{cases}$$

be the jth *ring set* for the center a_i. See Figure 3.1. It is easy to verify that every point in P lies in exactly one ring set, since no point of P can be in distance larger than $\beta n R$ from all the centers of $\mathcal{A}$. Therefore, these ring sets partition P into disjoint sets.

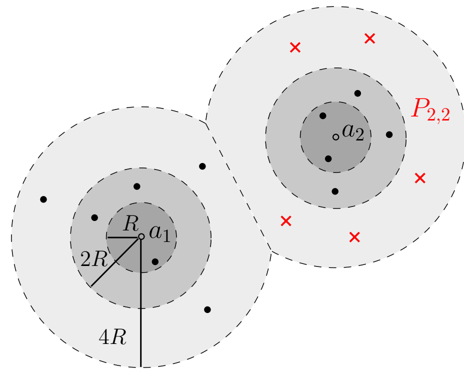


Figure 3.1: The ring sets. Here $\mathcal{A} = \{a_1, a_2\}$.

To compute the center set $\mathcal{A}$, in $O(nk)$ time, we use the algorithm of Indyk [Ind99], see Section 2.2 for details.

Step 2: Random sampling

Let

$$s = \left\lceil \frac{\mathbf{c}\beta^2}{\varepsilon^2} \left(k \ln n + \ln \frac{1}{\lambda} \right) \right\rceil, \tag{3.1}$$

where $\mathbf{c}$ is a sufficiently large constant. For $i = 1, \ldots m$ and $j = 0, \ldots, \phi$, if $|P_{i,j}| \leq s$ then set $\mathcal{S}_{i,j} = P_{i,j}$. Otherwise, randomly pick s points from $P_{i,j}$ independently and uniformly (with replacement), assign each point weight $|P_{i,j}|/s$, and let $\mathcal{S}_{i,j}$ be the resulting weighted sample. We assume that $|P_{i,j}|/s$ is an integer number.[1]

We claim that the set $\mathcal{S} = \cup_{i,j} \mathcal{S}_{i,j}$ is the desired (k, ε)-coreset of P.

3.2.2 Proof of correctness

Observation 3.2.1 *(i) For each $p \in P_{i,0}$, it holds $0 \leq \mathsf{d}(\mathcal{A}, p) \leq R$.*
(ii) For each $p \in P_{i,j}$, where $j \geq 1$, it holds $2^{j-1} R < \mathsf{d}(\mathcal{A}, p) \leq 2^j R$.

The following lemma is an easy variant of the result of Haussler [Hau92].

Lemma 3.2.2 *Let $M \geq 0$ and η be fixed constants, and $h(\cdot)$ be a function defined on a set V, such that $\eta \leq h(p) \leq \eta + M$, for all $p \in V$. Let $U = \{p_1, \ldots, p_s\}$ be a set of s samples drawn independently and uniformly from V, and let $\delta > 0$ be a parameter. If $s \geq (M^2/2\delta^2) \ln(2/\lambda)$, then $\mathbf{Pr}\left[\left| \frac{h(V)}{|V|} - \frac{h(U)}{|U|} \right| \geq \delta \right] \leq \lambda$, where $h(U) = \sum_{u \in U} h(u)$ and $h(V) = \sum_{v \in V} h(v)$.*

[1]This is a minor technicality that can be easily resolved. Indeed, if $|P_{i,j}|$ is not a multiple of s, we arbitrarily choose a set $Q_{i,j}$ of less than s points from $P_{i,j}$ such that $|P_{i,j} \setminus Q_{i,j}|$ is a multiple of s. Draw a set of s points from $P_{i,j} \setminus Q_{i,j}$ independently and uniformly, assign each sample point the weight $|P_{i,j} \setminus Q_{i,j}|/s$, and let $\mathcal{S}_{i,j}$ be the union of the weighted sample set (from $P_{i,j} \setminus Q_{i,j}$) and $Q_{i,j}$. It is easy to verify that $\mathbf{w}(\mathcal{S}_{i,j}) = \mathbf{w}(P_{i,j})$ and $|\mathcal{S}_{i,j}| \leq 2s$.

Lemma 3.2.3 *Let V be a set of points in a metric space (X, d), and $\lambda', \xi > 0$ be given parameters. Let U be a sample of $s' = \lceil \xi^{-2} \ln(2/\lambda') \rceil$ points picked from V, independently and uniformly, where each point of U is assigned weight $|V| / |U|$, such that $\mathbf{w}(U) = |V|$. For a fixed set C, we have that $|\nu(C, V) - \nu(C, U)| \leq \xi |V| \operatorname{diam}(V)$, with probability $\geq 1 - \lambda'$.*

Proof: Consider the function $h(v) = \mathsf{d}(C, v)$ defined over the points of V. By the triangle inequality, for every point $v \in V$, it holds

$$\mathsf{d}(C, V) \leq h(v) = \mathsf{d}(C, v) \leq \mathsf{d}(C, V) + \operatorname{diam}(V).$$

By Lemma 3.2.2, setting $\eta = \mathsf{d}(C, V)$, $M = \operatorname{diam}(V)$, and $\delta = \xi M$, we have that, for a sample U of size $s' = \lceil \xi^{-2} \ln(2/\lambda') \rceil \geq (M^2/2\delta^2) \ln(2/\lambda')$ from V, it holds

$$\mathbf{Pr}\left[\left| \frac{\sum_{v \in V} \mathsf{d}(C, v)}{|V|} - \frac{\sum_{u \in U} \mathsf{d}(C, u)}{|U|} \right| \geq \xi \operatorname{diam}(V) \right] = \mathbf{Pr}\left[\left| \frac{h(V)}{|V|} - \frac{h(U)}{|U|} \right| \geq \delta \right] \leq \lambda'.$$

This implies that

$$\begin{aligned} |\nu(C, V) - \nu(C, U)| &= |V| \cdot \left| \frac{\sum_{v \in V} \mathsf{d}(C, v)}{|V|} - \frac{\sum_{u \in U} \mathsf{d}(C, u) \mathbf{w}(u)}{|V|} \right| \\ &= |V| \cdot \left| \frac{\sum_{v \in V} \mathsf{d}(C, v)}{|V|} - \frac{\sum_{u \in U} \mathsf{d}(C, u)}{|U|} \right| \leq \xi |V| \operatorname{diam}(V), \end{aligned}$$

with probability $\geq 1 - \lambda'$, since $\mathbf{w}(u) = |V| / |U|$, for all $u \in U$. ■

Claim 3.2.4 *We have $\sum_{i,j} |P_{i,j}| 2^j R \leq 3\mathrm{opt}_\nu$ and $\sum_{i,j} |P_{i,j}| \operatorname{diam}(P_{i,j}) \leq 6\mathrm{opt}_\nu$, where $\mathrm{opt}_\nu = \mathrm{opt}_\nu(k, P)$.*

Proof: Let p be an arbitrary point in $P_{i,j}$. By Observation 3.2.1, we have $2^j R = R$ if $j = 0$, and $2^j R \leq 2\mathsf{d}(\mathcal{A}, p)$ if $j \geq 1$. Therefore, $2^j R \leq \max(2\mathsf{d}(\mathcal{A}, p), R) \leq 2\mathsf{d}(\mathcal{A}, p) + R$. Thus,

$$\begin{aligned} \sum_{i,j} |P_{i,j}| 2^j R &= \sum_{i,j} \sum_{p \in P_{i,j}} 2^j R \leq \sum_{i,j} \sum_{p \in P_{i,j}} (2\mathsf{d}(\mathcal{A}, p) + R) = \sum_{p \in P} (2\mathsf{d}(\mathcal{A}, p) + R) \\ &= 2\nu(\mathcal{A}, P) + |P| R \leq 2\beta \mathrm{opt}_\nu + \mathrm{opt}_\nu \leq 3\beta \mathrm{opt}_\nu, \end{aligned}$$

since $\nu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\nu$ and $|P| R = nR \leq \mathrm{opt}_\nu$. Now, since $\operatorname{diam}(P_{i,j}) \leq 2\big(2^j R\big)$, the above inequality also implies the second part of the claim. ■

Lemma 3.2.5 *For all sets $C \subseteq P$ of size at most k, it holds $|\nu(C, P) - \nu(C, \mathcal{S})| \leq \varepsilon \nu(C, P)$, with probability $\geq 1 - \lambda/2$.*

Proof: Fix an arbitrary set C of at most k centers. By Lemma 3.2.3, setting $\xi = \varepsilon/(6\beta)$ and $\lambda' = n^{-2k} \lambda/2$, it holds that

$$|\nu(C, P_{i,j}) - \nu(C, \mathcal{S}_{i,j})| \leq \frac{\varepsilon}{6\beta} |P_{i,j}| \operatorname{diam}(P_{i,j}),$$

with probability $\geq 1 - \lambda'$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$. Here, the sample required is of size $s' = \lceil \xi^{-2} \ln(2/\lambda') \rceil = \big\lceil (6\beta/\varepsilon)^2 \ln\big(4n^{2k}/\lambda\big) \big\rceil$. This is smaller than s, the actual number of points drawn from $P_{i,j}$, if $\mathbf{c}$ is sufficiently large, see Eq. (3.1). Now, by Claim 3.2.4, we have

$$\begin{aligned}|\nu(C,P)-\nu(C,\mathcal{S})| &\le \sum_{i,j}|\nu(C,P_{i,j})-\nu(C,\mathcal{S}_{i,j})| \\ &\le \frac{\varepsilon}{6\beta}\sum_{i,j}|P_{i,j}|\operatorname{diam}(P_{i,j}) \le \frac{\varepsilon}{6\beta}6\beta\mathrm{opt}_\nu \le \varepsilon\nu(C,P),\end{aligned}$$

and this holds with probability $\ge 1-m(\phi+1)\lambda'$.

There are at most $\sum_{i=1}^{k}\binom{n}{i} \le \sum_{i=1}^{k} n^i \le kn^k$ different ways to select a set C of at most k centers from P. As such, the above inequality holds for every set C of size at most k, with probability $\ge 1-kn^k m(\phi+1)\lambda' = 1-kn^k m(\phi+1)n^{-2k}\lambda/2 \ge 1-\lambda/2$. ■

Theorem 3.2.6 *Given a set P of n points in a metric space and parameters $1>\varepsilon>0$ and $\lambda>0$, one can compute a weighted set $\mathcal{S}$ of size $O\big(k\varepsilon^{-2}(k\log n+\log(1/\lambda))\log n\big)$, in $O(nk\log(1/\lambda))$ time, such that $\mathcal{S}$ is a (k,ε)-coreset of P for k-median clustering, with probability $\ge 1-\lambda$.*

If P is weighted, with total weight W, then the running time is $O(nk\log(1/\lambda)\log\log W)$, and the coreset size is $O\big(k\varepsilon^{-2}(k\log n+\log(1/\lambda))\log^2 W\big)$.

Proof: The algorithm is described in Section 3.2.1. By Theorem 2.2.5, the assumption that $\nu(\mathcal{A},P)\le \beta\mathrm{opt}_\nu(k,P)$ holds with probability $\ge 1-\lambda/2$. Therefore, by Lemma 3.2.5, it holds that $|\nu(C,P)-\nu(C,\mathcal{S})|\le \varepsilon\nu(C,P)$, for all sets $C\subseteq P$ of at most k centers, with probability $\ge 1-\lambda/2-\lambda/2 = 1-\lambda$. If P is unweighted then the size of the coreset $\mathcal{S}$ is $|\mathcal{S}| = O(m\phi s) = O\big(k\varepsilon^{-2}(k\log n+\log(1/\lambda))\log n\big)$, and if P is weighted then $|\mathcal{S}| = O(m\phi s) = O\big(k\varepsilon^{-2}(k\log n+\log(1/\lambda))\log^2 W\big)$.

The overall running time is dominated by the computation of the set $\mathcal{A}$. By Theorem 2.2.5, this takes $O(nk\log(1/\lambda))$ time if P is unweighted, and $O(nk\log(1/\lambda)\log\log W)$ time if P is weighted. ■

3.3 Coreset for Euclidean k-median clustering

In this section, we present an algorithm for computing coresets for Euclidean k-median clustering.

Definition 3.3.1 (Euclidean k-clustering.) Let P be a set of n points in $\mathbb{R}^d$. The *Euclidean k-median* (resp. *Euclidean k-means*) problem is to find a set of k centers $C\subseteq\mathbb{R}^d$ that minimizes the cost $\nu(C,P)$ (resp. $\mu(C,P)$), where the distance function d used is the usual Euclidean distance.

A weighted subset $\mathcal{S}\subseteq P$ is a (k,ε)-*coreset* of P for Euclidean k-median clustering, if $|\nu(C,\mathcal{S})-\nu(C,P)|\le \varepsilon\nu(C,P)$ for all sets C of at most k centers in $\mathbb{R}^d$. The (k,ε)-*coreset* of P for the Euclidean k-means clustering is defined similarly.

Note that, unlike the metric case, the center set under consideration can be any k-tuple of points in $\mathbb{R}^d$, which is not necessarily a subset of P.

3.3.1 The coreset construction

The algorithm is analogous to its metric variant. We point out the differences in the following. In the partitioning step, we use the same algorithm as described in Section 3.2.1. Let $\mathcal{A}=\{a_1,\ldots,a_m\}$ be a set of centers in $\mathbb{R}^d$ such that $\nu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\nu(k, P)$, where $m \leq \alpha k$ and $\alpha, \beta \geq 1$ (here, α and β are some constants). As before, $\mathcal{A}$ is computed using the algorithm of Indyk [Ind99], see Section 2.2. Let $R = \nu(\mathcal{A}, P)/(\beta n)$, and set

$$\phi = \left\lceil \lg\left(7\beta^2 n/\varepsilon\right) \right\rceil .$$

As in Section 3.2.1, we partition P into ring sets $P_{i,j}$. In the sampling step, we select s points from each ring set, where

$$s = \left\lceil \frac{\mathbf{c}'\beta^2}{\varepsilon^2} \left(k \ln(\alpha k) + k \ln \ln n + dk \ln \frac{\beta}{\varepsilon} + \ln \frac{1}{\lambda} \right) \right\rceil \tag{3.2}$$

and $\mathbf{c}'$ is a sufficiently large constant. Let $\mathcal{S}_{i,j}$ be the sample taken from $P_{i,j}$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$. We shall prove that $\mathcal{S} = \bigcup_{i,j} \mathcal{S}_{i,j}$ a (k, ε)-coreset.

Remark 3.3.2 A minor technicality here is that the algorithm of Indyk [Ind99] approximates the optimal metric k-median clustering (i.e., the *discrete* version of the Euclidean k-median clustering, where the centers must belong to P). Fortunately, the cost of the optimal Euclidean k-median clustering is at least half of the cost of the optimal discrete solution. As such, this algorithm still provides an $[O(1), O(1)]$-bi-criteria approximation to the optimal Euclidean k-median clustering.

3.3.2 Proof of correctness

The main challenge in proving the correctness of the above coreset construction is that there are infinite number of ways to select a set of at most k centers in $\mathbb{R}^d$ (versus only a finite number of ways to do so in the finite metric case). Thus, the arguments used in the proof of Lemma 3.2.5 are no longer valid. To circumvent this problem, we define a finite set $\mathcal{G}$ (note that the set $\mathcal{G}$ is used only in the analysis), and we will show that it is sufficient to prove correctness for center sets taken from $\mathcal{G}$. A similar (but weaker) notion of a witness set was used by Matoušek [Mat00].

Definition 3.3.3 Let $\mathcal{U}$ be the union of "huge" balls centered at the points of $\mathcal{A}$. Formally, $\mathcal{U} = \bigcup_{i=1}^m \mathrm{ball}(a_i, 2^\phi R)$, where $a_i \in \mathcal{A}$. For $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$, let

$$L_{i,j} = \begin{cases} \mathrm{ball}(a_i, R) & j = 0 \\ \mathrm{ball}(a_i, 2^j R) \setminus \mathrm{ball}(a_i, 2^{j-1} R) & j \geq 1. \end{cases}$$

We use an axis-parallel grid with side length $\varrho_j = 2^j \varepsilon R / (\mathsf{b}\, \beta \sqrt{d})$ to partition $L_{i,j}$ into cells, where $\mathsf{b} = 50$. Inside each grid cell of $L_{i,j}$, pick an arbitrary point (say, the center of the cell) as its *representative point*. Let $\mathcal{G}_{i,j}$ denote the set of representative points for $L_{i,j}$, and let $\mathcal{G} = \bigcup_{i,j} \mathcal{G}_{i,j}$.

Claim 3.3.4 *We have* $\ln|\mathcal{G}| = O(\log(\alpha k) + \log \log n + d \log(\beta/\varepsilon))$.

Proof: Fix $L_{i,j}$, and consider a cell $\mathsf{c}_{i,j}$ in the grid partitioning $L_{i,j}$. The volume of $\mathsf{c}_{i,j}$ is

$$\mathrm{vol}(\mathsf{c}_{i,j}) = (\varrho_j)^d = \left(\frac{2^j R \varepsilon}{\mathsf{b}\, \beta \sqrt{d}} \right)^d .$$

Note that the distance from any point of $\mathsf{c_{i,j}}$ to a_i is at most $2^j R + \mathrm{diam}(\mathsf{c_{i,j}}) < 2^{j+1}R$, which implies $\mathsf{c_{i,j}} \subseteq B_{i,j} = \mathrm{ball}(a_i, 2^{j+1}R)$. Therefore, the number of cells inside $L_{i,j}$, denoted by $\omega_{i,j}$, is at most $\mathrm{vol}(B_{i,j})/\mathrm{vol}(\mathsf{c_{i,j}})$. Applying the formula of the volume of a ball in $\mathbb{R}^d$ to $B_{i,j}$, we obtain

$$\mathrm{vol}(B_{i,j}) = \frac{\pi^{d/2}(2^{j+1}R)^d}{\Gamma(d/2+1)},$$

where $\Gamma(\cdot)$ is the gamma function (which is an extension of the factorial function). In particular, $\Gamma(d/2+1) \geq d'!$, where $d' = \lfloor d/2 \rfloor$, for $d \geq 4$. Since $n! \geq (n/e)^n$, it holds that $\Gamma(d/2+1) \geq d'! \geq (d'/e)^{d'} \geq (d/4e)^{d/2}$. This implies

$$\begin{aligned}
\omega_{i,j} &= \frac{\mathrm{vol}(B_{i,j})}{\mathrm{vol}(\mathsf{c_{i,j}})} \leq \frac{\pi^{d/2}(2^{j+1}R)^d}{\Gamma(d/2+1)} \left(\frac{\mathsf{b}\,\beta\sqrt{d}}{2^j R\varepsilon} \right)^d \\
&\leq \frac{\pi^{d/2}(2^{j+1}R)^d}{(d/4e)^{d/2}} \cdot \frac{(\mathsf{b}\,\beta)^d d^{d/2}}{(2^j R\varepsilon)^d} \leq \left(\frac{2\,\mathsf{b}\,\beta}{\varepsilon} \right)^d \left(\frac{\pi d}{d/4e} \right)^{d/2} = \left(\frac{\mathsf{b}'\beta}{\varepsilon} \right)^d,
\end{aligned}$$

where $\mathsf{b}' = 4\sqrt{\pi e}\,\mathsf{b} < 16\,\mathsf{b}$. Now, the size of $\mathcal{G}$ is

$$|\mathcal{G}| \leq \sum_{i,j} \omega_{i,j} \leq m(\phi+1) \left(\frac{\mathsf{b}'\beta}{\varepsilon} \right)^d \leq \alpha k \left(\lg \frac{7\beta^2 n}{\varepsilon} + 1 \right) \left(\frac{\mathsf{b}'\beta}{\varepsilon} \right)^d,$$

and thus,

$$\ln|\mathcal{G}| \leq \ln(\alpha k) + \ln\left(\lg \frac{7\beta^2 n}{\varepsilon} + 1 \right) + d \ln \frac{\mathsf{b}'\beta}{\varepsilon} = O\left(\log(\alpha k) + \log\log n + d \log \frac{\beta}{\varepsilon} \right),$$

as claimed. ■

Lemma 3.3.5 *With probability $\geq 1 - \lambda/2$, for all sets C' of at most k centers chosen from $\mathcal{G}$, it holds that $|\nu(C', P) - \nu(C', \mathcal{S})| \leq (\varepsilon/5)\,\nu(C', P)$.*

Proof: The argument follows the proof of Lemma 3.2.5. As in Lemma 3.2.5, we need the sample to work for all subsets of size at most k of $\mathcal{G}$, and the number of such subsets is at most $k|\mathcal{G}|^k$. Therefore, to achieve confidence $1 - \lambda/2$, set $\xi = \varepsilon/(\mathsf{b}\,\beta)$ and $\lambda' = |\mathcal{G}|^{-2k}\lambda/2$. The required sample size is $s' = \lceil \xi^{-2} \ln(2/\lambda') \rceil$. By Claim 3.3.4, it holds

$$\begin{aligned}
s' &\leq 2\xi^{-2} \ln \frac{2}{\lambda'} = 2\left(\frac{\mathsf{b}\,\beta}{\varepsilon} \right)^2 \ln\left(\frac{4|\mathcal{G}|^{2k}}{\lambda} \right) \leq 2\left(\frac{\mathsf{b}\,\beta}{\varepsilon} \right)^2 \left(2k \ln|\mathcal{G}| + \ln \frac{4}{\lambda} \right) \\
&= O\left(\frac{\beta^2}{\varepsilon^2} \left(k \log\log n + dk \log \frac{\beta}{\varepsilon} + k \log(\alpha k) + \log \frac{1}{\lambda} \right) \right).
\end{aligned}$$

This is smaller than s, the number of samples used, if $\mathbf{c}'$ is sufficiently large, see Eq. (3.2). ■

Claim 3.3.6 *(i) It holds that $\nu(\mathcal{A}, \mathcal{S}) \leq 3\beta \mathrm{opt}_\nu$, where* $\mathrm{opt}_\nu = \mathrm{opt}_\nu(k, P)$.

(ii) For any set C of centers, it holds $|\nu(C, P) - \nu(C, \mathcal{S})| \leq 6\beta \mathrm{opt}_\nu$.

Proof: Consider a ring set $P_{i,j}$ and its corresponding weighted sample set $\mathcal{S}_{i,j}$. Because $\mathbf{w}(\mathcal{S}_{i,j}) = |P_{i,j}|$, there exists a map $f : P_{i,j} \to \mathcal{S}_{i,j}$ such that $\left|f^{-1}(q)\right| = \mathbf{w}(q)$, $\forall q \in \mathcal{S}_{i,j}$.

(i) By Claim 3.2.4, we have

$$\nu(\mathcal{A}, \mathcal{S}) = \sum_{i,j} \nu(\mathcal{A}, \mathcal{S}_{i,j}) = \sum_{i,j} \sum_{p \in P_{i,j}} \mathsf{d}(\mathcal{A}, f(p)) \leq \sum_{i,j} \sum_{p \in P_{i,j}} 2^j R = \sum_{i,j} |P_{i,j}| \, 2^j R \leq 3\beta \mathrm{opt}_\nu,$$

since $\mathsf{d}(\mathcal{A}, f(p)) \leq 2^j R$, for any $p \in P_{i,j}$.

(ii) Let p be an arbitrary point in $P_{i,j}$. By the triangle inequality, it holds that $\mathsf{d}(C, f(p)) + \mathsf{d}(f(p), p) \geq \mathsf{d}(C, p)$ and $\mathsf{d}(C, p) + \mathsf{d}(p, f(p)) \geq \mathsf{d}(C, f(p))$, and as such, we have that

$$|\mathsf{d}(C, p) - \mathsf{d}(C, f(p))| \leq \mathsf{d}(p, f(p)) \leq \mathrm{diam}(P_{i,j}).$$

Therefore, by Claim 3.2.4,

$$\begin{aligned} |\nu(C, P) - \nu(C, \mathcal{S})| &\leq \sum_{i,j} |\nu(C, P_{i,j}) - \nu(C, \mathcal{S}_{i,j})| = \sum_{i,j} \left| \sum_{p \in P_{i,j}} (\mathsf{d}(C, p) - \mathsf{d}(C, f(p))) \right| \\ &\leq \sum_{i,j} \sum_{p \in P_{i,j}} \mathrm{diam}(P_{i,j}) = \sum_{i,j} |P_{i,j}| \, \mathrm{diam}(P_{i,j}) \leq 6\beta \mathrm{opt}_\nu. \end{aligned}$$

■

In the following, let $C \subseteq \mathbb{R}^d$ be an arbitrary set of at most k centers. We need to prove that $|\nu(C, \mathcal{S}) - \nu(C, P)| \leq \varepsilon \nu(C, P)$. Recall that $\mathcal{U}$ is a union of balls centered at the points of $\mathcal{A}$, see Definition 3.3.3.

Lemma 3.3.7 *For $0 < \varepsilon < 1$, if there exists a center $c \in C$ and a point $p \in P$ such that c is outside $\mathcal{U}$ and $\mathsf{d}(C, p) = \|cp\|$, then $|\nu(C, \mathcal{S}) - \nu(C, P)| \leq \varepsilon \nu(C, P)$.*

Proof: Let a_p be the nearest center to p in $\mathcal{A}$. We have $\|a_p p\| \leq \nu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\nu$, where $\mathrm{opt}_\nu = \mathrm{opt}_\nu(k, P)$. In addition, since c is outside $\mathcal{U}$, it holds that $\|c a_p\| \geq \mathsf{d}(\mathcal{A}, c) \geq 2^\phi R \geq 7\beta^2 n R/\varepsilon \geq 7\beta \mathrm{opt}_\nu/\varepsilon$. By the triangle inequality, we thus have

$$\nu(C, P) \geq \nu(C, p) = \|cp\| \geq \|c a_p\| - \|a_p p\| \geq 7\beta \mathrm{opt}_\nu/\varepsilon - \beta \mathrm{opt}_\nu \geq 6\beta \mathrm{opt}_\nu/\varepsilon.$$

Now, by Claim 3.3.6 (ii), we have $|\nu(C, \mathcal{S}) - \nu(C, P)| \leq 6\beta \mathrm{opt}_\nu \leq \varepsilon \nu(C, P)$. ■

The above lemma implies that we can assume that $C \subseteq \mathcal{U}$ (since otherwise, the set $\mathcal{S}$ is the required coreset). Therefore, suppose that $C = \{c_1, \ldots, c_h\}$, where $h \leq k$. Let $C' = \{c'_1, \ldots, c'_h\}$, where $c'_t \in \mathcal{G}$ is the representative point of the cell containing c_t, for $t = 1, \ldots, h$.

Lemma 3.3.8 *If $C \subseteq \mathcal{U}$ and $|C| \leq k$, then $|\nu(C, q) - \nu(C', q)| \leq \dfrac{\varepsilon}{\mathsf{b}\beta}(2\nu(C, q) + 2\nu(\mathcal{A}, q) + R)$, for any $q \in P$.*

Proof: Let c_i and c'_j be the nearest centers to q in C and C', respectively. Namely, $\nu(C, q) = \|c_i q\|$ and $\nu(C', q) = \left\|c'_j q\right\|$. We consider the case when $\nu(C, q) \leq \nu(C', q)$, as the other case is similar. By the triangle inequality, it holds that

$$|\nu(C,q) - \nu(C',q)| = \nu(C',q) - \nu(C,q) = \left\|c'_j q\right\| - \|c_i q\| \leq \|c'_i q\| - \|c_i q\| \leq \|c'_i c_i\|,$$

since $\left\|c'_j q\right\| \leq \|c'_i q\|$. If $\mathsf{d}(\mathcal{A}, c_i) \leq R$, then

$$\|c_i c'_i\| \leq \sqrt{d} \cdot \frac{\varepsilon R}{\mathsf{b}\,\beta\sqrt{d}} = \frac{\varepsilon R}{\mathsf{b}\,\beta},$$

and this implies the required bound. Otherwise, we have $2^{t-1}R < \mathsf{d}(\mathcal{A}, c_i) \leq 2^t R$, for some $t \geq 1$. Then c_i and c'_i are inside a cell of side length $2^t \varepsilon R/(\mathsf{b}\,\beta\sqrt{d})$, and as such,

$$\begin{aligned} \|c_i c'_i\| &\leq \sqrt{d} \cdot \frac{2^t \varepsilon R}{\mathsf{b}\,\beta\sqrt{d}} = \frac{2\varepsilon}{\mathsf{b}\,\beta} \cdot 2^{t-1}R < \frac{2\varepsilon}{\mathsf{b}\,\beta}\mathsf{d}(\mathcal{A}, c_i) \\ &\leq \frac{2\varepsilon}{\mathsf{b}\,\beta}(\|c_i q\| + \mathsf{d}(\mathcal{A}, q)) = \frac{2\varepsilon}{\mathsf{b}\,\beta}(\nu(C,q) + \nu(\mathcal{A}, q)), \end{aligned}$$

by the triangle inequality. ■

Lemma 3.3.9 *If $C \subseteq \mathcal{U}$ and $|C| \leq k$, then*

(i) $|\nu(C,P) - \nu(C',P)| \leq (\varepsilon/10)\nu(C,P)$, and

(ii) $|\nu(C,\mathcal{S}) - \nu(C',\mathcal{S})| \leq (\varepsilon/2)\nu(C,P)$.

Proof: (i) Recall that $\mathsf{b} = 50$. Summing up the inequality of Lemma 3.3.8 over all the points of P, we obtain

$$\begin{aligned} |\nu(C,P) - \nu(C',P)| &\leq \frac{\varepsilon}{\mathsf{b}\,\beta}(2\nu(C,P) + 2\nu(\mathcal{A},P) + nR) \\ &\leq \frac{\varepsilon}{50\beta}(2\nu(C,P) + 2\beta\nu(C,P) + \nu(C,P)) \leq \frac{\varepsilon}{10}\nu(C,P), \end{aligned}$$

since $\nu(\mathcal{A},P) \leq \beta \mathrm{opt}_\nu \leq \beta\nu(C,P)$ and $nR \leq \mathrm{opt}_\nu \leq \nu(C,P)$, where $\mathrm{opt}_\nu = \mathrm{opt}_\nu(k,P)$.

(ii) Summing up the inequality of Lemma 3.3.8 over all the weighted points of $\mathcal{S} \subseteq P$, we obtain

$$\begin{aligned} |\nu(C,\mathcal{S}) - \nu(C',\mathcal{S})| &\leq \frac{\varepsilon}{\mathsf{b}\,\beta}(2\nu(C,\mathcal{S}) + 2\nu(\mathcal{A},\mathcal{S}) + nR) \leq \frac{\varepsilon}{\mathsf{b}\,\beta}(2\nu(C,P) + 18\beta\mathrm{opt}_\nu + \mathrm{opt}_\nu) \\ &\leq \frac{\varepsilon}{50\beta}(2\nu(C,P) + 18\beta\nu(C,P) + \nu(C,P)) \leq \frac{\varepsilon}{2}\nu(C,P), \end{aligned}$$

since $\nu(C,\mathcal{S}) \leq \nu(C,P) + 6\beta\mathrm{opt}_\nu$, by Claim 3.3.6 (ii), and $\nu(\mathcal{A},\mathcal{S}) \leq 3\beta\mathrm{opt}_\nu$, by Claim 3.3.6 (i). ■

Lemma 3.3.10 *For $0 < \varepsilon < 1$, and for any $C \subseteq \mathcal{U}$ such that $|C| \leq k$, it holds that*

$$|\nu(C,P) - \nu(C,\mathcal{S})| \leq \varepsilon\nu(C,P),$$

and this holds with probability $\geq 1 - \lambda/2$.

Proof: By Lemma 3.3.9 (i), we have $\nu(C', P) \leq \nu(C, P) + (\varepsilon/10)\nu(C, P) \leq (11/10)\nu(C, P)$. Therefore, by Lemma 3.3.5, we have

$$|\nu(C', P) - \nu(C', \mathcal{S})| \leq \frac{\varepsilon}{5}\nu(C', P) \leq \frac{\varepsilon}{5} \cdot \frac{11}{10}\nu(C, P) \leq \frac{\varepsilon}{4}\nu(C, P).$$

Now, by Lemma 3.3.9, it holds

$$\begin{aligned}
|\nu(C, P) - \nu(C, \mathcal{S})| &\leq |\nu(C, P) - \nu(C', P)| + |\nu(C', P) - \nu(C', \mathcal{S})| + |\nu(C', \mathcal{S}) - \nu(C, \mathcal{S})| \\
&\leq \frac{\varepsilon}{10}\nu(C, P) + \frac{\varepsilon}{4}\nu(C, P) + \frac{\varepsilon}{2}\nu(C, P) < \varepsilon\nu(C, P),
\end{aligned}$$

and this holds with probability at least $\geq 1 - \lambda/2$, since Lemma 3.3.5 holds with probability $\geq 1 - \lambda/2$. ■

Putting the above together implies the following.

Theorem 3.3.11 *Given a set P of n points in $\mathbb{R}^d$ and parameters $1 > \varepsilon > 0$ and $\lambda > 0$, one can compute a set $\mathcal{S}$, of size*

$$O\left(\frac{k \log n}{\varepsilon^2}\left(dk \log \frac{1}{\varepsilon} + k \log k + k \log \log n + \log \frac{1}{\lambda}\right)\right),$$

in $O(ndk \log(1/\lambda))$ time. The set $\mathcal{S}$ is a (k, ε)-coreset of P for k-median clustering, with probability $\geq 1 - \lambda$.

If P is weighted, with total weight W, then the running time is $O(ndk \log(1/\lambda) \log \log W)$, and the coreset size is

$$O\left(\frac{k \log^2 W}{\varepsilon^2}\left(dk \log \frac{1}{\varepsilon} + k \log k + k \log \log W + \log \frac{1}{\lambda}\right)\right).$$

3.4 Coreset for k-means clustering

In this section, we present algorithms for computing coresets for k-means clustering.

3.4.1 Coreset for metric k-means clustering

Assume that $\mathcal{A} = \{a_1, \ldots, a_m\} \subseteq P$ satisfies $\mu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\mu(k, P)$, where $m \leq \alpha k$ and $\alpha, \beta \geq 1$. Let $R = \sqrt{\mu(\mathcal{A}, P)/(\beta n)}$ be a lower bound estimate of the average radius of the optimal k-means clustering. As before, we compute the set $\mathcal{A}$ using the algorithm of Indyk [Ind99], see Section 2.2.

We construct ring sets $P_{i,j}$ and combine the samples $\mathcal{S}_{i,j}$ from all ring sets, as in the metric k-median case. The correctness proof proceeds similarly as in the k-median case. For the sake of completeness, in the following, we prove the required lemmas that imply the correctness of the algorithm.

Lemma 3.4.1 *Let V be a set of points in a metric space (X, d), and $\lambda', \xi > 0$ be given parameters. Let U be a sample of $s' = \lceil 4\xi^{-2} \ln(2/\lambda') \rceil$ points picked from V independently and uniformly, where each point of U is assigned weight $|V| / |U|$, such that $\mathbf{w}(U) = |V|$. For a fixed set C, we have that*

$$|\mu(C, V) - \mu(C, U)| \leq \xi |V| \left[(\mathsf{d}(C, V))^2 + (\mathrm{diam}(V))^2\right],$$

with probability $\geq 1 - \lambda'$.

Proof: Consider the function $h(v) = (\mathsf{d}(C, v))^2$ defined over the points of V. Observe that for a point $v \in V$,

$$0 \leq h(v) = (\mathsf{d}(C, v))^2 \leq \left(\mathsf{d}(C, V) + \mathrm{diam}(V)\right)^2 \leq 2\left(\mathsf{d}(C, V)\right)^2 + 2\left(\mathrm{diam}(V)\right)^2.$$

The remainder of the proof is similar to the proof of Lemma 3.2.3, and we omit the easy modifications. ■

Claim 3.4.2 *For any fixed set C such that $|C| \leq k$, we have*

$$\sum_{i,j} |P_{i,j}| \left[(\mathsf{d}(C, P_{i,j}))^2 + (\mathrm{diam}(P_{i,j}))^2 \right] \leq 21\beta\mu(C, P).$$

Proof: Let p be an arbitrary point in $P_{i,j}$. We have $2^j R \leq 2\mathsf{d}(\mathcal{A}, p)$ if $j \geq 1$, and $2^j R = R$ if $j = 0$. Therefore, since $\mathrm{diam}(P_{i,j}) \leq 2^{j+1} R$, it holds that

$$(\mathrm{diam}(P_{i,j}))^2 \leq (2^{j+1} R)^2 = 4(2^j R)^2 \leq 4\left[(2\mathsf{d}(\mathcal{A}, p))^2 + R^2\right] = 16(\mathsf{d}(\mathcal{A}, p))^2 + 4R^2.$$

Furthermore, we have $(\mathsf{d}(C, P_{i,j}))^2 \leq (\mathsf{d}(C, p))^2$. As such,

$$\begin{aligned}
\sum_{i,j} |P_{i,j}| \left[(\mathsf{d}(C, P_{i,j}))^2 + (\mathrm{diam}(P_{i,j}))^2 \right] &= \sum_{i,j} \sum_{p \in P_{i,j}} \left[(\mathsf{d}(C, P_{i,j}))^2 + (\mathrm{diam}(P_{i,j}))^2 \right] \\
&\leq \sum_{i,j} \sum_{p \in P_{i,j}} \left[(\mathsf{d}(C, p))^2 + 16(\mathsf{d}(\mathcal{A}, p))^2 + 4R^2 \right] \\
&= \mu(C, P) + 16\mu(\mathcal{A}, P) + 4\,|P|\,R^2 \leq 21\beta\mu(C, P),
\end{aligned}$$

since $\mu(\mathcal{A}, P) \leq \beta \mathrm{opt}_\mu(k, P) \leq \beta\mu(C, P)$ and $|P|\,R^2 = nR^2 \leq \mathrm{opt}_\mu(k, P) \leq \mu(C, P)$. ■

The following lemma is similar to Lemma 3.2.5, and we omit the proof.

Lemma 3.4.3 *For all sets $C \subseteq P$ of size at most k, it holds that $|\mu(C, P) - \mu(C, \mathcal{S})| \leq \varepsilon\mu(C, P)$, with probability $\geq 1 - \lambda/2$.*

Continuing in a similar fashion to Section 3.2.2, we get the following result.

Theorem 3.4.4 *Given a set P of n points in a metric space and parameters $1 > \varepsilon > 0$ and $\lambda > 0$, one can compute a weighted set $\mathcal{S}$ of size $O(k\varepsilon^{-2} \log n (k \log n + \log(1/\lambda)))$, in $O(nk \log(1/\lambda))$ time, such that $\mathcal{S}$ is a (k, ε)-coreset of P for k-means clustering, with probability $\geq 1 - \lambda$.*

If P is weighted, with total weight W, then the running time is $O(nk \log(1/\lambda) \log\log W)$, and the coreset size is $O(k\varepsilon^{-2} \log^2 W (k \log n + \log(1/\lambda)))$.

3.4.2 Coreset for Euclidean k-means clustering

For the Euclidean k-means clustering, the construction and the correctness proofs are similar to those in the Euclidean k-median case. As such, we omit the easy but tedious details, and only state the result below (which corresponds to Theorem 3.3.11).

Theorem 3.4.5 *Given a set P of n points in $\mathbb{R}^d$ and parameters $1 > \varepsilon > 0$ and $\lambda > 0$, one can compute a weighted set $\mathcal{S}$ of size*

$$O\left(\frac{k \log n}{\varepsilon^2}\left(dk \log \frac{1}{\varepsilon} + k \log k + k \log \log n + \log \frac{1}{\lambda}\right)\right),$$

in $O(ndk \log(1/\lambda))$ time, such that $\mathcal{S}$ is a (k, ε)-coreset of P for k-means clustering, with probability $\geq 1 - \lambda$.

If P is weighted, with total weight W, then the running time is $O(ndk \log(1/\lambda) \log \log W)$, and the coreset size is

$$O\left(\frac{k \log^2 W}{\varepsilon^2}\left(dk \log \frac{1}{\varepsilon} + k \log k + k \log \log W + \log \frac{1}{\lambda}\right)\right).$$

3.5 Applications

In this section, we provide applications for the (k, ε)-coreset constructions described in Section 3.2, Section 3.3, and Section 3.4. We can plug the coresets into any k-clustering algorithm that works for a weighted point set.

3.5.1 Faster clustering algorithms

In the metric spaces, we plug in the local search algorithm of Arya *et al.* [AGK$^+$01] into our machinery. Specifically, we compute a constant factor bi-criteria approximation for the optimal solution using `FastCluster` (described in Section 2.2). Next, we apply the coreset construction of Theorem 3.2.6 to compute a (k, ε)-coreset. Now, using the local search algorithm [AGK$^+$01] yields the required approximation. We summarize:

Theorem 3.5.1 *Given a set P of n points in a metric space, one can compute a $(10 + \varepsilon)$-approximate k-median clustering of P in $O(nk + k^7 \varepsilon^{-4} \log^5 n)$ time, with constant probability of success.*

In $\mathbb{R}^d$, we use the same algorithm with the twist that we use in the final stage the $(1+\varepsilon)$-approximate algorithm of Kumar *et al.* [KSS04, KSS05] (instead of the local search algorithm in the metric case). A simple extension of their algorithm to work on a weighted input yields a $(1+\varepsilon)$-approximation algorithm for k-median clustering of P in $\mathbb{R}^d$. In particular, let $T(n, m)$ be the running time of their algorithm on the (k, ε)-coreset, where n is the total weight of the coreset and m is the number of centers, the recurrence of $T(n, m)$ is

$$T(n, m) = O(u(k, \varepsilon))T(n, m - 1) + T(n/2, m) + O((c(k, \varepsilon) + u(k, \varepsilon))d),$$

where $u(k, \varepsilon) = O(2^{(k/\varepsilon)^{O(1)}})$ and $c(k, \varepsilon)$ is the size of the (k, ε)-coreset, see [KSS04, KSS05] for details. It is not difficult to show that $T(n, k) = O(d 2^{(k/\varepsilon)^{O(1)}} c(k, \varepsilon) k n^\sigma)$, for any fixed constant $\sigma > 0$.

Theorem 3.5.2 *Given a set P of n points in $\mathbb{R}^d$, one can compute a $(1+\varepsilon)$-approximation to the optimal k-median (or k-means) clustering of P in time $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^\sigma)$, with constant probability, for any fixed $\sigma > 0$.*

3.5.2 Streaming

Coresets were used to design approximation algorithms in the streaming model [HM04a, AHV04]. In particular, Har-Peled and Mazumdar [HM04a] used coresets to develop approximation algorithms for k-clustering in the insertion-only streaming model. The randomized coreset construction described in this chapter can also be used in the streaming model using the same techniques. In the following, we adapt the algorithm of Har-Peled and Mazumdar [HM04a] to our randomized coresets.

The algorithm of Har-Peled and Mazumda is based on the standard dynamization technique of Bentley and Saxe [BS80] and the following observation.

Observation 3.5.3 *(i) If $\mathcal{S}_1$ and $\mathcal{S}_2$ are the (k,ε)-coresets for disjoint sets P_1 and P_2 respectively, then $\mathcal{S}_1 \cup \mathcal{S}_2$ is a (k,ε)-coreset for $P_1 \cup P_2$.*

(ii) If $\mathcal{S}_1$ is (k,ε)-coreset for $\mathcal{S}_2$ and $\mathcal{S}_2$ is a (k,δ)-coreset for $\mathcal{S}_3$, then $\mathcal{S}_1$ is a $(k,(1+\varepsilon)(1+\delta)-1)$-coreset for $\mathcal{S}_3$.

Suppose that a sequence of points $p_1, p_2, \ldots$ in $\mathbb{R}^d$ arrive one by one in a stream. We want to compute a coreset for the k-clustering of the points that arrived so far, and the result should be correct with probability $\geq 1-\lambda$, where λ is a pre-specified confidence parameter.

We use buckets $B_0, B_1, \ldots$ to store the points. The capacity of bucket B_0 is M, where M is a parameter to be specified shortly, and the capacity of bucket B_i is $2^{i-1}M$, for $i \geq 1$. We will keep the invariant that B_i is either full or empty, for $i \geq 1$. When p_m arrives, we insert p_m into B_0. If B_0 has less than M points, then we are done. Otherwise, let $t \geq 1$ be the smallest index such that B_t is empty, merge all the points of $B_0, \ldots, B_{t-1}$ into B_t. Here, B_t is *triggered* by p_m. (After B_t is triggered by p_m, the buckets $B_0, \ldots, B_{t-1}$ become empty and B_t becomes full.)

However, we can not afford (space-wise) to keep every point in the buckets in the streaming model. Instead, we maintain a coreset Q_i for each bucket B_i. Q_0 is B_0 itself, and whenever B_t is triggered by p_m, let Q_t be a (k,ρ_t)-coreset of $\bigcup_{i=0}^{t-1} Q_i$ with confidence parameter $\lambda_m = \lambda/m^2$, where $\rho_t = \varepsilon/(\mathsf{b}\,(t+1)^2)$ and b is a sufficiently large constant. Let $Q = \bigcup_{i\geq 0} Q_i$.

Claim 3.5.4 *The set Q is a (k,ε)-coreset of the points received so far, with probability $\geq 1-\lambda$.*

Proof: Recall that $\rho_r = \varepsilon/(\mathsf{b}\,(r+1)^2)$. It is easy to verify that $\prod_{l=0}^{r}(1+\rho_l) \leq 1+\varepsilon$ if b is sufficiently large. On the other hand, Q_r is a $(k, \prod_{l=0}^{r}(1+\rho_l)-1)$-coreset of B_r, by applying Observation 3.5.3 repeatedly. Therefore, Q_r is a (k,ε)-coreset of the points in B_r, and $Q = \bigcup_{i\geq 0} Q_i$ is a (k,ε)-coreset of the points in $\bigcup_{i\geq 0} B_i$. That is, Q is a (k,ε)-coreset of the points received so far.

When we process the newly arrived point p_m, our computation may fail with probability $\leq \lambda_m = \lambda/m^2$ whenever we compute a coreset with confidence parameter λ_m. When p_m arrives, where $m \geq M$, it may trigger at most one coreset computations. As such, overall the algorithm may fail with probability $\leq \sum_{i=M}^{n} \lambda_i = \sum_{i=M}^{n} (\lambda/i^2) \leq \lambda$, for $M \geq 2$. ■

Set $M = \lceil k^2\varepsilon^{-2}d \rceil$ and assume that we have received n points so far. Note that $|Q_0| \leq M$. For $i = 1, \ldots, \lceil \lg n \rceil$, Q_i has a total weight $2^{i-1}M$ (if it is not empty) and it is generated as a $\left(k, \varepsilon/(\mathsf{b}\, i^2)\right)$-coreset of $\bigcup_{j=0}^{i-1} Q_j$ with confidence parameter at least λ/n^2. By Theorem 3.3.11, we have that

$$|Q_i| = O\left(\frac{ki^4(i+\log M)^2}{\varepsilon^2}\left(dk\log\frac{i^2}{\varepsilon} + k\log k + k\log(i+\log M) + \log\frac{n^2}{\lambda}\right)\right).$$

If $\lambda = 1/\operatorname{poly}(n)$, then the total storage requirement is

$$M + \sum_{i=1}^{\lceil \lg n \rceil} |Q_i| = O\big(dk^2 \varepsilon^{-2} \log^8 n\big) \, .$$

To analyze the update time of the data structure, observe that the amortized time dealing with Q_0 is constant, and Q_i is constructed after every $2^{i-1}M$ insertions are made, for $i = 1, \ldots, \lceil \lg n \rceil$. Therefore by Theorem 3.3.11, the amortized time spent for an update is

$$O\left(\sum_{i=1}^{\lceil \lg n \rceil} \frac{\sum_{j=0}^{i-1} |Q_j|}{2^{i-1}M} dk \big(\log \log \big(2^{i-1}M\big)\big) \log \frac{n^2}{\lambda} \right) = O\left(dk \big(\log^2 n\big) \operatorname{polylog}\left(\frac{dk}{\varepsilon}\right)\right) .$$

We summarize:

Theorem 3.5.5 *Given a stream P of n points in $\mathbb{R}^d$ and $\varepsilon > 0$, one can maintain a (k, ε)-coreset for k-median (or k-means) clustering efficiently for the points seen so far. The coreset is correct with high probability. The space used is $O(dk^2\varepsilon^{-2}\log^8 n)$, and the amortized update time is $O(dk \operatorname{polylog}(ndk/\varepsilon))$.*

3.6 Conclusions

In this chapter, we used sampling techniques to extract a small (k, ε)-coreset for k-clustering in both metric spaces and high dimensional Euclidean spaces. Such a coreset construction for metric spaces was not known before. In high dimensional Euclidean spaces, this is the first construction with polynomial dependency on the dimension. The coreset can be used to obtain fast approximation algorithms for the k-median and k-means problems. It is especially useful in the streaming model of computation, where the small storage space is desired. In particular, we provide the first streaming clustering algorithm that has space complexity with polynomial dependency on the dimension.

In addition, the small coreset leads to a $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^\sigma)$-time $(1+\varepsilon)$-approximation algorithm to the optimal k-clustering in $\mathbb{R}^d$, which succeeds with constant probability, for any fixed constant $\sigma > 0$. This improves over the work of Kumar *et al.* [KSS04, KSS05]. This result, together with the low dimensional result of Har-Peled and Mazumdar [HM04a], indicates, surprisingly, that the expensive part in computing k-clustering in $\mathbb{R}^d$ is answering nearest neighbor queries (this is the $O(ndk)$ term in the running time). In particular, a slight speedup can be achieved by using a fast data-structure for approximate nearest neighbor, see [IM98].

In light of the recent result of Har-Peled and Kushal [HK05] (see also [ES04]), which constructed low dimensional coreset of size independent of n (but exponential in the dimension), it is natural to ask if one can construct a coreset of size with polynomial dependency on the dimension and with no dependency on n. We leave this as open problem for further research. A more intriguing possibility is that one can construct coresets of size independent of the dimension altogether, as was done in the min-enclosing ball case [BC03].

Chapter 4

Approximation Algorithm for k-median with m Outliers

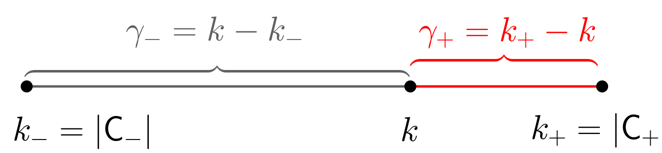
In this chapter, we present the first polynomial time constant factor approximation algorithm for the *k-median clustering with outliers problem.* In Section 4.2, we present the algorithm. In Section 4.3, we provide the intuition why the algorithm works, and prove some key properties. In Section 4.4 and Section 4.5, we prove the correctness of the algorithm.

4.1 Preliminaries

In the remainder of the chapter, we consider the problem instance $\mathsf{MO}(k, \mathsf{P}, m)$, where P is a given set of n points. For technical reasons, we assume that the distances between all pairs of points in P are distinct, and the spread of P is polynomially bounded, in particular, $d_{max}/d_{min} = O(n^2)$, where d_{max} and d_{min} are the maximal and minimal inter-point distances in P, respectively. One can slightly perturb the distance function d so that it fulfills those requirements. See Section 2.5 for details.

4.1.1 The Lagrangian approach

Let FloAlg denote the algorithm provided for FLO by Charikar *et al.* [CKMN01]. (In fact, the constant approximation factors provided by the algorithm of Mahdian [Mah04] are slightly better, but this does not affect our results substantially.) See Section 2.3.



Consider $\mathsf{FLO}(z, \mathsf{P}, m)$. When $z = 0$, the algorithm FloAlg opens all the facilities, and when $z = nd_{max}$, it opens only a single facility. We perform a binary search on the interval $[0, nd_{max}]$ to find z_- and z_+ such that the algorithm opens $k_- \leq k$ and $k_+ \geq k+1$ facilities for $\mathsf{FLO}(z_-, \mathsf{P}, m)$ and $\mathsf{FLO}(z_+, \mathsf{P}, m)$, respectively, and moreover, $|z_- - z_+| \leq d_{min}/n^2$ (this can be done in $O(\log n)$ steps, since the spread of P is polynomially bounded). Let C_- and C_+ be the facility sets computed by the algorithm for z_- and z_+, respectively. Here $|\mathsf{C}_-| = k_-$ and $|\mathsf{C}_+| = k_+$.

Let $\gamma_- = k - k_-$ and $\gamma_+ = k_+ - k$. We have $\gamma_- \geq 0$ and $\gamma_+ \geq 1$, since $k_- \leq k$ and $k_+ \geq k + 1$. Also, we have $\gamma_- + \gamma_+ = k_+ - k_-$.

4.1.2 A modified point set P^{w}

Let $M_+ = \mathbf{N}_{n-m}(\mathsf{C}_+, \mathsf{P})$ be the set of the $n - m$ points in P closest to C_+. Snap each point of M_+ to its nearest neighbor in C_+, and let P^{w} denote the resulting multiset.

Definition 4.1.1 (Heavy point.) A point p is *heavy* if $p \in \mathsf{C}_+$. Its weight, denoted by $\mathsf{w}(p)$, is the number of points in M_+ served by p. Given two heavy points p and q, if $\mathsf{w}(p) > \mathsf{w}(q)$ then p is *heavier* than q. A point p is *light* if $p \in \mathsf{P} \setminus M_+$ (that is, p is one of the m outliers in the solution induced by C_+). A light point has weight one.

In P^{w}, there are exactly k_+ heavy points and m light points, and the points in $M_+ \setminus \mathsf{C}_+$ (i.e., those with weight zero) are neither heavy nor light, as they are being collapsed to the points of C_+.

Note that P^{w} is the set $\mathsf{C}_+ \cup (\mathsf{P} \setminus M_+)$ with appropriate weights associated with the points. As such, the *size* of P^{w} is $|\mathsf{P}^{\mathsf{w}}|_{\mathsf{w}} = \mathsf{w}(\mathsf{P}) = n$, and the number of distinct points in P^{w} is $k_+ + m$. The multiset P^{w} can be thought of as a *coreset* of P, which is roughly a coarse representation of the original set P. (The interested reader is referred to [HM04b] for definition.) Informally, the cost of clustering P^{w} by any set C (of k facilities) is roughly the cost of clustering P by C. In fact, up to a constant factor, we can even restrict C to lie in P^{w} and still get a constant factor approximation to the optimal cost. In particular, we have the following folklore lemma.

Lemma 4.1.2 *For any $C \in \mathsf{P}$, there exists $C' \in \mathsf{P}^{\mathsf{w}}$ such that*

$$\mathsf{A}_m(C, \mathsf{P}) = \Theta\Big(\mathsf{A}_m(C', \mathsf{P}) + \mathsf{opt}\Big) = \Theta\Big(\mathsf{A}_m(C', \mathsf{P}^{\mathsf{w}}) + \mathsf{opt}\Big).$$

Definition 4.1.3 (Include, exclude, and partly-include.) Given a heavy point p and a set $Q \subseteq \mathsf{P}^{\mathsf{w}}$, if p occurs $\mathsf{w}(p)$ times in Q then it *includes* p, if p does not appear in Q then it *excludes* p, and otherwise, it *partly-includes* p.

Definition 4.1.4 (The set $\mathcal{X}$.) Let $\mathcal{X}'$ be the set of $n-m$ points in P^{w} closest to C_-. Since all distances (between distinct points) in P^{w} are distinct, there might be (only) one heavy point, say q, which is partly-included in $\mathcal{X}'$. In this case, we remove all copies of q from $\mathcal{X}'$ and let $\mathcal{X}$ be the resulting set, otherwise, set $\mathcal{X} = \mathcal{X}'$.

For a set $B \subseteq \mathsf{P}^{\mathsf{w}}$, let $h_{\mathsf{w}}(B)$ denote the number of distinct heavy points in B, and $l_{\mathsf{w}}(B)$ denote the number of light points in B (note that each light point appears exactly once in P^{w}, and as such the light points are distinct).

Definition 4.1.5 (Mass, cost, and benefit.) If $l_{\mathsf{w}}(\mathcal{X}) = 0$ then let $\xi = 0$. Otherwise, let

$$\xi = \frac{k_+ - h_{\mathsf{w}}(\mathcal{X}) - 1}{l_{\mathsf{w}}(\mathcal{X})}. \tag{4.1}$$

For a point $p \in \mathcal{X}$, let $\mathrm{cost}(p) = \nu(\mathsf{C}_-, p)$. The *mass* of p, denoted by $\mathrm{mass}(p)$, is ξ if p is light, and $1/\mathsf{w}(p)$ otherwise. For a set $B \subseteq \mathcal{X}$ of points, let $\mathrm{mass}(B) = \sum_{p \in B} \mathrm{mass}(p)$ and $\mathrm{cost}(B) = \sum_{p \in B} \mathrm{cost}(p)$, and the *benefit* of B is $\mathrm{ben}(B) = \mathrm{mass}(B) - 1$.

4.1.3 The local search method

We shall reduce MO to the penalty k-median with m outliers problem (PMO), which is defined below, and apply the local search method to PMO.

$\mathbf{N}_{n-m}(C, P)$	The set of $n - m$ points in P closest to C.
$\nu(C, P)$	Cost of connecting the points in P to their nearest facilities in C.
$\mathsf{A}_m(C, P)$	$\nu(C, M)$, where M consists of the $n - m$ points in P closest to C.
$\mathsf{MO}(k, P, m)$	An instance of k-median with m outliers, with objective to compute $C \subseteq P$ minimizing $\mathsf{A}_m(C, P)$.
$\mathcal{A}_m(C, P, \varrho)$	Cost of connecting M to C, where M consists of the $n - m$ points in P closest to C, and each point $p \in M$ pays a cost of $\min(\varrho, \mathsf{d}(C, p))$.
$\mathsf{PMO}(k, P, \varrho, m)$	An instance of penalty k-median with m outliers, with objective to compute $C \subseteq P$ minimizing $\mathcal{A}_m(C, P, \varrho)$.
$\mathrm{opt}_{\mathsf{mo}}(k, P, m)$	the cost of the optimal solution to $\mathsf{MO}(k, P, m)$.
$\mathrm{opt}_{\mathsf{pmo}}(k, P, \varrho, m)$	the cost of the optimal solution to $\mathsf{PMO}(k, P, \varrho, m)$.
opt	$\mathrm{opt}_{\mathsf{mo}}(k, \mathsf{P}, m)$, the cost of the optimal solution to $\mathsf{MO}(k, \mathsf{P}, m)$.
$\mathsf{opt}^{\mathsf{w}}$	$\mathrm{opt}_{\mathsf{mo}}(k, \mathsf{P}^{\mathsf{w}}, m)$, the cost of the optimal solution to $\mathsf{MO}(k, \mathsf{P}^{\mathsf{w}}, m)$.

Figure 4.1: Notations.

In the PMO problem, we are allowed to exclude more than m outliers, but every such additional outlier incurs a penalty. Equivalently, given a set P of n points, a set C of facilities, and a penalty parameter $\varrho > 0$, let

$$\mathcal{A}_m(C, P, \varrho) = \sum_{p \in \mathbf{N}_{n-m}(C,P)} \min\Big(\mathsf{d}(C, p), \varrho\Big)$$

denote the cost of PMO clustering P with m outliers and penalty ϱ, where $\mathbf{N}_{n-m}(C, P)$ is the set of $n-m$ points in P closest to C. Namely, we assign $\mathbf{N}_{n-m}(C, P)$ to C. Every point $p \in \mathbf{N}_{n-m}(C, P)$ pays a connection cost, which is the distance $\mathsf{d}(C, p)$ capped by the *penalty* ϱ.

Definition 4.1.6 (Penalty k-median with m outliers.) Let $\mathsf{PMO}(k, P, \varrho, m)$ denote an instance of penalty k-median with m outliers, consisting of an integer $k \geq 1$, a set P of points, a penalty parameter $\varrho > 0$, and $m \geq 0$. The objective is to compute a set $C \subseteq P$ of k facilities minimizing the cost $\mathcal{A}_m(C, P, \varrho)$. Let $\mathrm{opt}_{\mathsf{pmo}}(k, P, \varrho, m)$ denote the cost of the optimal solution.

Observe that the problem $\mathsf{PMO}(k, P, \varrho, m)$ is a relaxation of $\mathsf{MO}(k, P, m)$. In particular, for $\varrho = \infty$, we have $\mathcal{A}_m(C, P, \varrho) = \mathsf{A}_m(C, P)$.

Definition 4.1.7 (Neighbor facility sets.) Given a set $C \subseteq \mathsf{P}^{\mathsf{w}}$ of k facilities, let

$$\mathtt{N}(C) = \{C\} \cup \{C - q' + q'' \mid q' \in C, q'' \in \mathsf{P}^{\mathsf{w}} \setminus C\}$$

denote the *neighbor facility sets* of C, where $C - q' + q'' = (C \setminus \{q'\}) \cup \{q''\}$.

Definition 4.1.8 (The sets H and $\mathcal{H}$.) Recall that there are $|\mathsf{C}_+| = k_+ \geq k + 1$ heavy points in P^{w}. Let H consists of the k heaviest among them, and

$$\mathcal{H} = \{C \mid C \subseteq \mathsf{P}^{\mathsf{w}}, C \text{ contains at least } k-1 \text{ heavy points, and } |C| = k\}.$$

4.2 The algorithm

The input is the set P and parameters k and m. The algorithm uses binary search over the range $[0, nd_{max}]$ to find z_- and z_+ such that $|z_- - z_+| \leq d_{min}/n^2$, and the sets $\mathsf{C}_- = \textsc{FloAlg}(z_-, \mathsf{P}, m)$ and $\mathsf{C}_+ = \textsc{FloAlg}(z_+, \mathsf{P}, m)$ satisfy $|\mathsf{C}_-| \leq k$ and $|\mathsf{C}_+| \geq k+1$. (Here, $\textsc{FloAlg}$ is used to make the decision in the binary search.) See Section 4.1.1 for details. Next, it computes a multiset P^{w} by collapsing the clusters (of P) induced by C_+ into their respective facilities, see Section 4.1.2. The algorithm checks if $\gamma_+ = k_+ - k \geq 2$, and if so, it uses $\textsc{ClusterSparse}$, described below in Section 4.2.1, to compute the desired solution C. Otherwise, $\gamma_+ = 1$, and the algorithm uses $\textsc{ClusterDense}$, described in Section 4.2.2.

4.2.1 The algorithm ClusterSparse for the case $\gamma_+ \geq 2$

We shall compute a set $\mathsf{C} \subseteq \mathsf{C}_- \cup \mathsf{C}_+$ such that $|\mathsf{C}| = k$ and it is the required solution.

Suppose $\mathsf{C}_- = \{f_1, \ldots, f_{k_-}\}$, and let $\mathcal{X}_i$ be the set of points of $\mathcal{X}$ that are nearest to f_i, for $i = 1, \ldots, k_-$. Assume, without loss of generality, that $\mathrm{ben}(\mathcal{X}_1), \ldots, \mathrm{ben}(\mathcal{X}_\alpha) > 0$, for some $1 \leq \alpha \leq k_-$, and $\mathrm{ben}(\mathcal{X}_{\alpha+1}), \ldots, \mathrm{ben}\big(\mathcal{X}_{k_-}\big) \leq 0$, and furthermore,

$$\frac{\mathrm{cost}(\mathcal{X}_1)}{\mathrm{ben}(\mathcal{X}_1)} \leq \cdots \leq \frac{\mathrm{cost}(\mathcal{X}_\alpha)}{\mathrm{ben}(\mathcal{X}_\alpha)}.$$

Let k' be the index satisfying $\sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) < \gamma_+ \leq \sum_{t=1}^{k'} \mathrm{ben}(\mathcal{X}_t)$, where $k' \leq \alpha$. Construct a set C of k facilities as follows.

(i) Let $\overline{\mathcal{Y}} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{k'-1}, \mathcal{Y}_{k'}\}$. The set $\mathcal{Y}_{k'}$ is generated greedily from $\mathcal{X}_{k'}$ by repeatedly picking the point p in $\mathcal{X}_{k'}$ (that has not been added yet) with the smallest $\mathrm{cost}(p)/\mathrm{mass}(p)$ value. Here, if p is heavy, we add in all its copies. We repeat this till

$$\mathrm{BEN}\big(\overline{\mathcal{Y}}\big) = \sum_{B \in \overline{\mathcal{Y}}} \mathrm{ben}(B) \in [\gamma_+, \gamma_+ + 1) \tag{4.2}$$

for the first time.

(ii) Let $J \subseteq \mathsf{C}_+$ be the set of $k - \big|\overline{\mathcal{Y}}\big| = k - k'$ heaviest points not included in $\mathcal{Y} = \bigcup \overline{\mathcal{Y}}$.

(iii) Return $\mathsf{C} = \{f_1, \ldots, f_{k'}\} \cup J$.

4.2.2 The algorithm ClusterDense for the case $\gamma_+ = 1$

The algorithm $\textsc{ClusterDense}(k, \mathsf{P}^{\mathsf{w}}, m)$ is presented in Figure 4.2 (a). Its input consists of the point set P^{w}, C_+, and integers k and m, and it returns the desired approximation. The procedure $\textsc{ClusterDense}$ uses $\textsc{LocalSearch}$, depicted in Figure 4.2 (b). Here, the set C_- is not used by the algorithm, and C_+ is used to derive the sets H and $\mathcal{H}$, see Definition 4.1.8.

Intuitively, $\textsc{ClusterDense}$ works by generating a set of candidate facility sets, among which at least one is more expensive than the optimal solution by only a constant factor. Therefore, the cheapest solution among the candidates generated provides the required approximation.

Algorithm ClusterDense$(k, \mathsf{P}^{\mathsf{w}}, m)$
    $i \leftarrow 0$
    $\varrho_0 \leftarrow d_{min}/10$
    $B_0 \leftarrow \mathsf{H}$
    $\Delta_0 \leftarrow \Delta(B_0, \mathsf{P}^{\mathsf{w}}, \varrho_0, m)$.
    while $\Delta_i > 0$ **do**
        $i \leftarrow i + 1$
        $\varrho_i \leftarrow 3\varrho_{i-1}$
        $B_i \leftarrow$ LocalSearch$(B_{i-1}, \mathsf{P}^{\mathsf{w}}, \varrho_i)$
        $\Delta_i \leftarrow \Delta(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$.
    $\mathcal{X} \leftarrow \mathcal{H} \cup \bigcup_{t=0}^{i} \mathtt{N}(B_t)$
    return $\operatorname{argmin}_{C \in \mathcal{X}} \mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}})$.

(a)

Algorithm LocalSearch$(B, \mathsf{P}^{\mathsf{w}}, \varrho)$
    while $\exists B' \in \mathtt{N}(B) \cup \{\mathsf{H}\}$ such that
        $\mathcal{Z}(B') < \mathcal{Z}(B) - \dfrac{\varrho}{3}$ **do**
        $B \leftarrow B'$
    return B

(b)

Figure 4.2: (a) A successive local search algorithm for $\mathsf{MO}(k, \mathsf{P}^{\mathsf{w}}, m)$. Here, $\Delta(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$ is the number of points that pay the penalty ϱ_i in the PMO clustering induced by B_i. Formally, this is the number of points in $\mathbf{N}_{n-m}(B_i, \mathsf{P}^{\mathsf{w}})$ that are in distance $\geq \varrho_i$ from B_i, see Section 4.1.3. (b) Here, $\mathcal{Z}(B')$ refers to $\mathcal{A}_m(B', \mathsf{P}^{\mathsf{w}}, \varrho)$, and $\mathcal{Z}(B)$ refers to $\mathcal{A}_m(B, \mathsf{P}^{\mathsf{w}}, \varrho)$.

4.2.3 The result

We have the following result.

Theorem 4.2.1 *Given a set* P *of* n *points, integral parameters* $k \geq 1$ *and* $m \geq 0$*, one can compute, in* $O(k^2(k+m)^2 n^3 \log n)$ *time, a set* $\mathsf{C} \subseteq \mathsf{P}$ *of* k *facilities such that* $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) = O(\mathsf{opt})$*, where* $\mathsf{opt} = \mathrm{opt}_{\mathsf{mo}}(k, \mathsf{P}, m)$.

The rest of the chapter is dedicated to proving Theorem 4.2.1. In particular, it is implied by Lemma 4.4.15 and Lemma 4.5.8.

4.3 Intuition and correctness

4.3.1 Intuition

We handle the two cases $\gamma_+ \geq 2$ and $\gamma_+ = 1$ separately, because a key claim (see Claim 4.3.4) used in bounding the cost of C_- works only for the case $\gamma_+ \geq 2$, see also Remark 4.3.5. Moreover, the analysis of the local search method does not hold in the case $\gamma_+ \geq 2$, see Lemma 4.5.5.

Intuition for ClusterSparse ($\gamma_+ \geq 2$). In the clustering of P^{w} induced by C_+, every heavy point itself is a cluster (recall that the total weight of heavy points is $n-m$). ClusterSparse needs to "pack" these k_+ clusters (i.e., heavy points) into k clusters, with the help of C_-. Note that $\mathcal{X}_1, \ldots, \mathcal{X}_{k_-}$ are the k_- clusters in the clustering of $\mathcal{X}$ induced by C_-, and intuitively, consider $\mathcal{X}_1, \ldots, \mathcal{X}_{k_-}$ as a MO clustering of P^{w} (recall that $|\mathcal{X}|_{\mathsf{w}}$ is roughly $n - m$). To do the packing, we assign a mass of one to (all copies of) each heavy point. Intuitively, the mass of $\mathcal{X}_i$ is the (fractional) number of heavy points in $\mathcal{X}_i$. The mass of $\mathcal{X}_i$ may be fractional, since it might contain light points. The mass of a light point p (i.e., ξ) is the fraction of the heavy points that are "ejected" from $\mathcal{X}$ because of p (if p is included in $\mathcal{X}$, then some heavy points must have been excluded by $\mathcal{X}$). Naturally, we would like to use $\mathcal{X}_i$ with maximum mass, since it packs the largest number of (fractional) heavy points into a single new cluster. In fact, a cluster $\mathcal{X}_i$ with mass one or less does not help us in this merging process (since $\mathcal{X}_i$ would use one facility on its own). In particular, we are mainly interested in the (added) benefit of $\mathcal{X}_i$, namely $\mathrm{ben}(\mathcal{X}_i) = \mathrm{mass}(\mathcal{X}_i) - 1$. Furthermore, great benefit with prohibitive cost is of little use for us. As such, we sort the $\mathcal{X}_i$s by their *return*, namely $\mathrm{cost}(\mathcal{X}_i) / \mathrm{ben}(\mathcal{X}_i)$. Next, we pick as many of them as necessary so that we can add the remaining (uncovered) heavy points as clusters to the solution, and still use only k facilities.

Intuition for ClusterDense ($\gamma_+ = 1$). Here, we reduce the k-median with m outliers problem (MO) to the penalty k-median with m outliers (PMO). The objective of MO is to compute C minimizing $\mathsf{A}_m(\mathsf{C}, \mathsf{P}^{\mathsf{w}})$, while PMO aims to minimize $\mathcal{A}_m(\mathsf{C}, \mathsf{P}^{\mathsf{w}}, \varrho)$. Observe that those two cost functions are the same when the penalty parameter ϱ is sufficiently large. Therefore, if we can obtain a constant factor approximation solution for PMO (with a large penalty parameter), then we are done (because it is also a constant factor approximation for MO). Furthermore, when the penalty is small enough (i.e., less than the minimal inter-point distance), the optimal solution to PMO is easy to compute — it is just H, the set of the k heaviest points in P^{w}. Now, we start with a (very) small penalty parameter, and gradually increase the penalty parameter by "doubling" it in each round. Because the penalty parameter increases "slowly", and the solution computed from each round is used as the starting point for the next round, we argue that the solution of LocalSearch tracks the optimal solution cost. This implies that, when the penalty parameter becomes large enough, we have the required approximation. More formally, let $\overline{\omega}_i$ be the cost of the optimal solution to PMO in the ith round, and let ω_i be the cost of the corresponding LocalSearch solution (in the same round). Roughly, since $\omega_i - \omega_{i-1} = O(\overline{\omega}_i - \overline{\omega}_{i-1})$, for every $i \geq 1$, we have $\omega_i = O(\overline{\omega}_i)$. In particular, for i sufficiently large, we obtain the required approximation.

4.3.2 Correctness

Observation 4.3.1 *Let P be a set of n points, $C \subseteq P$ be a set of facilities, and M' be a set of at least $n - m$ points in P, see Definition 2.3.1. It holds that $\mathsf{A}_m(C, P) \leq \nu(C, M')$.*

Lemma 4.3.2 *Given a set P of points and non-negative parameters m and z, let C be the facility set computed by* FloAlg *for $\mathsf{FLO}(z, P, m)$. It holds that, for any $k \geq 1$,*

$$\mathsf{A}_m(C, P) \leq 3\mathrm{opt}_{\mathsf{mo}}(k, P, m) + 3z(k - |C| + 1).$$

Proof: We have $\mathrm{opt}_{\mathsf{flo}}(z, P, m) \leq \mathrm{opt}_{\mathsf{mo}}(k, P, m) + zk$, for any $k \geq 1$, as $\mathrm{opt}_{\mathsf{mo}}(k, P, m) + zk$ is the FLO cost of serving P using the k optimal facilities realizing $\mathrm{opt}_{\mathsf{mo}}(k, P, m)$. Now, it follows from Theorem 2.3.10 that

$$\begin{aligned} \mathsf{A}_m(C, P) &\leq 3\mathrm{opt}_{\mathsf{flo}}(z, P, m) - 3z(|C| - 1) \leq 3\Big(\mathrm{opt}_{\mathsf{mo}}(k, P, m) + zk\Big) - 3z(|C| - 1) \\ &= 3\mathrm{opt}_{\mathsf{mo}}(k, P, m) + 3z(k - |C| + 1). \end{aligned}$$

■

The following is motivated by the work of Jain and Vazirani [JV01] on k-median clustering. Conceptually, they merge C_- and C_+ by using the fractional solution

$$\mathsf{C}^* = \frac{\gamma_+}{\gamma_- + \gamma_+}\mathsf{C}_- + \frac{\gamma_-}{\gamma_- + \gamma_+}\mathsf{C}_+. \tag{4.3}$$

Here, a facility in C^* is now assigned a fractional weight and the total weight of C^* is k. This provides a convex combination of the two solutions into a single solution. Next, Jain and Vazirani use a random merging procedure to realize an integral facility having (roughly) the cost of C^* (in expectation). Furthermore, the cost of C_+ is $O(OPT)$ and the cost of C_- can be bounded by $O\left(\frac{\gamma_- + \gamma_+}{\gamma_+}OPT\right)$, where OPT is the cost of the optimal solution. Plugging this into Eq. (4.3) yields the required approximation.

However, our situation here is more subtle, since we have different outlier sets associated with the two solutions that we need to merge. In particular, there does not seem to be an easy way to adapt their algorithm to this problem.

Claim 4.3.3 *We have* $\mathsf{A}_m(\mathsf{C}_+, \mathsf{P}) \leq 3\mathsf{opt}$*, where* $\mathsf{opt} = \mathrm{opt}_{\mathsf{mo}}(k, \mathsf{P}, m)$.

Proof: Since $\gamma_+ = k_+ - k = |\mathsf{C}_+| - k$, it holds that, by Lemma 4.3.2,

$$\mathsf{A}_m(\mathsf{C}_+, \mathsf{P}) \leq 3\mathsf{opt} + 3z_+(k - |\mathsf{C}_+| + 1) = 3\mathsf{opt} + 3z_+(1 - \gamma_+). \tag{4.4}$$

Note that $z_+ \geq 0$ and $\gamma_+ \geq 1$, as such, we have $\mathsf{A}_m(\mathsf{C}_+, \mathsf{P}) \leq 3\mathsf{opt}$. ■

Claim 4.3.4 *If* $\gamma_+ \geq 2$*, then* $\mathsf{A}_m(\mathsf{C}_-, \mathsf{P}) \leq 9\,\frac{\gamma_- + \gamma_+}{\gamma_+}\mathsf{opt}$.

Proof: We first bound z_+. By Eq. (4.4), we have

$$3z_+(\gamma_+ - 1) \leq 3\mathsf{opt} - \mathsf{A}_m(\mathsf{C}_+, \mathsf{P}) \leq 3\mathsf{opt},$$

which implies $z_+ \leq \frac{\mathsf{opt}}{\gamma_+ - 1}$. Since $z_- \leq z_+ + \frac{d_{min}}{n^2}$ and $d_{min} \leq \mathsf{opt}$, it follows that

$$\begin{aligned} z_-(\gamma_- + 1) &\leq \left(z_+ + \frac{d_{min}}{n^2}\right)(\gamma_- + 1) \leq \left(\frac{\mathsf{opt}}{\gamma_+ - 1} + \frac{\mathsf{opt}}{n^2}\right)(\gamma_- + 1) \\ &= \left(\frac{\gamma_- + 1}{\gamma_+ - 1} + \frac{\gamma_- + 1}{n^2}\right)\mathsf{opt} \leq \frac{\gamma_- + 2}{\gamma_+ - 1}\mathsf{opt}, \end{aligned}$$

since $\frac{\gamma_- + 1}{n^2} \leq \frac{1}{\gamma_+ - 1}$. Now, by Lemma 4.3.2, we obtain

$$\begin{aligned} \mathsf{A}_m(\mathsf{C}_-, \mathsf{P}) &\leq 3\mathsf{opt} + 3z_-(k - |\mathsf{C}_-| + 1) = 3\mathsf{opt} + 3z_-(\gamma_- + 1) \\ &\leq \left(3 + 3\frac{\gamma_- + 2}{\gamma_+ - 1}\right)\mathsf{opt} = 3\,\frac{\gamma_+ + \gamma_- + 1}{\gamma_+ - 1}\mathsf{opt}. \end{aligned}$$

We have $\gamma_+ - 1 \geq \frac{\gamma_+}{2}$ since $\gamma_+ \geq 2$, and $\gamma_+ + \gamma_- + 1 \leq \frac{3}{2}(\gamma_+ + \gamma_-)$ since $\gamma_+ + \gamma_- \geq \gamma_+ \geq 2$. As such,

$\dfrac{\gamma_+ + \gamma_- + 1}{\gamma_+ - 1} \le 3\dfrac{\gamma_+ + \gamma_-}{\gamma_+}$, implying the claim. ■

Remark 4.3.5 If $\gamma_+ = 1$ then z_+ cannot be bounded by using Lemma 4.3.2, as done in Claim 4.3.4. In fact, z_+ may be arbitrarily large compared to opt in this case. As such, a similar claim to Claim 4.3.4 does not hold here, and the convex combination in Eq. $(4.3)_{\text{p39}}$ is not necessarily a constant approximation for MO. This is the reason why we cannot apply ClusterSparse in this case.

If $\gamma_+ \ge 2$ and $\dfrac{\gamma_- + \gamma_+}{\gamma_+} = O(1)$ then, by Claim 4.3.4, the set C_- is the required approximation (since $|\mathsf{C}_-| = k_- \le k$). For example, if $k_+ \ge 2k$, then $\dfrac{\gamma_- + \gamma_+}{\gamma_+} \le 2$, and as such $\mathsf{A}_m(\mathsf{C}_-, \mathsf{P}) \le 18\mathsf{opt}$. If $\gamma_+ \ge 2$ and $\gamma_- \le u$, for some $u \ge 0$, then we have $\dfrac{\gamma_- + \gamma_+}{\gamma_+} \le 1 + u$ and as such $\mathsf{A}_m(\mathsf{C}_-, \mathsf{P}) \le (9 + 9u)\mathsf{opt}$. In particular, for a fixed u, the solution C_- yields the required constant factor approximation. Henceforth, we assume that $k_+ < 2k$. Furthermore, if $\gamma_+ \ge 2$, then we assume that $\gamma_- > 3$.

Lemma 4.3.6 *(i) For $C \subseteq \mathsf{P}$, we have that $|\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) - \mathsf{A}_m(C, \mathsf{P})| \le 3\mathsf{opt}$.*

(ii) If $\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) \le \gamma\, \mathsf{opt}^{\mathsf{w}}$, for some $\gamma \ge 1$, then $\mathsf{A}_m(C, \mathsf{P}) \le (4\gamma + 3)\mathsf{opt}$.

Proof: (i) We will prove that $|\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) - \mathsf{A}_m(C, \mathsf{P})| \le \mathsf{A}_m(\mathsf{C}_+, \mathsf{P})$. Because, by Claim 4.3.3, $\mathsf{A}_m(\mathsf{C}_+, \mathsf{P}) \le 3\mathsf{opt}$, this implies the claim. In the following, we focus on the case when $\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) \le \mathsf{A}_m(C, \mathsf{P})$, since the other case is similar.

Let Q and Q' be the (multi)sets of $n-m$ nearest points to C in P^{w} and P, respectively. By the definition of ϕ and w (see Section 4.1.2), there exists a set $Q'' \subseteq \mathsf{P}$ of $n - m$ points such that $\{\phi(p) \mid p \in Q''\} = Q$. Therefore,

$$\begin{aligned} \nu(C, Q'') - \nu(C, Q) &= \sum_{p \in Q''} (\mathsf{d}(p, C) - \mathsf{d}(\phi(p), C)) \le \sum_{p \in \mathsf{P}} |\mathsf{d}(p, C) - \mathsf{d}(\phi(p), C)| \\ &\le \sum_{p \in \mathsf{P}} |\mathsf{d}(p, \phi(p))| = \mathsf{A}_m(\mathsf{C}_+, \mathsf{P}). \end{aligned}$$

In addition, we have $\nu(C, Q') - \nu(C, Q'') \le 0$, since Q' is the set of $n - m$ nearest points to C in P. It thus follows that

$$\begin{aligned} |\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) - \mathsf{A}_m(C, \mathsf{P})| &= \mathsf{A}_m(C, \mathsf{P}) - \mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) = \nu(C, Q') - \nu(C, Q) \\ &= (\nu(C, Q') - \nu(C, Q'')) + (\nu(C, Q'') - \nu(C, Q)) \\ &\le 0 + \mathsf{A}_m(\mathsf{C}_+, \mathsf{P}). \end{aligned}$$

(ii) Suppose that C_o is an optimal solution for $\mathsf{MO}(k, \mathsf{P}, m)$, namely $|C_o| = k$ and $\mathsf{A}_m(C_o, \mathsf{P}) = \mathsf{opt}$. Then, by (i), we have

$$\mathsf{opt}^{\mathsf{w}} \le \mathsf{A}_m(C_o, \mathsf{P}^{\mathsf{w}}) \le 3\mathsf{opt} + \mathsf{A}_m(C_o, \mathsf{P}) = 4\mathsf{opt}.$$

It follows that $\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) \le \gamma \mathsf{opt}^{\mathsf{w}} \le 4\gamma \mathsf{opt}$, which implies by (i) the claim. ■

The following corollary is implied by Claim 4.3.4 and Lemma 4.3.6 (i).

Corollary 4.3.7 *If $\gamma_+ \ge 2$ then $\mathsf{A}_m(\mathsf{C}_-, \mathsf{P}^{\mathsf{w}}) \le \left(3 + 9\dfrac{\gamma_- + \gamma_+}{\gamma_+}\right) \mathsf{opt}$.*

4.4 Correctness of ClusterSparse ($\gamma_+ \geq 2$)

In this section, we show that, for the case $\gamma_+ \geq 2$, ClusterSparse computes a solution C such that $|\mathsf{C}| = k$ and $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) \leq 39\mathsf{opt}$. Here, we assume that $\gamma_- \geq 3$, see Remark 4.3.5.

Let $Z = \mathcal{Y} \cup J^{\mathsf{w}}$, where $\mathcal{Y}$ and J are the sets constructed in the step (i) and step (ii) of ClusterSparse, respectively. The cost $\nu(\mathsf{C}, Z)$ is equal to $\mathrm{cost}(\mathcal{Y})$, and it is in turn $O\Big(\frac{\gamma_+}{\gamma_- + \gamma_+}\mathrm{cost}(\mathcal{X})\Big)$, see Lemma 4.4.7 below. Moreover, Corollary 4.3.7 implies that $\mathrm{cost}(\mathcal{X}) = O\Big(\frac{\gamma_- + \gamma_+}{\gamma_+}\mathsf{opt}\Big)$, and combining these inequalities yields

$$\nu(\mathsf{C}, Z) = \mathrm{cost}(\mathcal{Y}) = O\bigg(\frac{\gamma_+}{\gamma_- + \gamma_+}\mathrm{cost}(\mathcal{X})\bigg) = O\bigg(\frac{\gamma_+}{\gamma_- + \gamma_+} \cdot \frac{\gamma_- + \gamma_+}{\gamma_+}\mathsf{opt}\bigg) = O(\mathsf{opt}).$$

We are not quite done yet, as we have to argue that the size of Z is at least $n - m$, see Lemma 4.4.14. This claim is intuitively implied by $\mathrm{BEN}(\overline{\mathcal{Y}}) \geq \gamma_+$ (see Eq. $(4.2)_{\mathrm{p36}}$) but the proof is tedious, and we defer it to Appendix 4.4.3.

4.4.1 ClusterSparse is sound

In this section, we show that all the steps of the algorithm succeed. Indeed, Claim 4.4.3 below proves that k', used in step (i) of ClusterSparse, does exist. Also, in step (i), we always have $\mathrm{mass}(p) > 0$, as the mass of any point in $\mathcal{X}$ is positive. In step (ii) of ClusterSparse, we have $k' \leq k_- < k$, and furthermore, Claim 4.4.4 below implies that at least $k - k'$ heavy points are excluded by $\mathcal{Y}$, thus guaranteeing that step (ii) succeeds.

Observation 4.4.1 *(i) All heavy points are either included or excluded by $\mathcal{X}$.*

(ii) If $l_{\mathsf{w}}(\mathcal{X}) = 0$ then $h_{\mathsf{w}}(\mathcal{X}) = k_+$, and if $l_{\mathsf{w}}(\mathcal{X}) > 0$ then $h_{\mathsf{w}}(\mathcal{X}) \leq k_+ - 1$.

(iii) We have $\xi \geq 0$, see Eq. $(4.1)_{p34}$. Moreover, for a set $B \subseteq \mathcal{X}$, we have

$$\mathrm{mass}(B) = h_{\mathsf{w}}(B) + \xi \cdot l_{\mathsf{w}}(B). \tag{4.5}$$

Claim 4.4.2 *(i) If $l_{\mathsf{w}}(\mathcal{X}) = 0$ then* $\mathrm{mass}(\mathcal{X}) = k_+$*, and if $l_{\mathsf{w}}(\mathcal{X}) > 0$ then* $\mathrm{mass}(\mathcal{X}) = k_+ - 1$.

(ii) $\sum_{i=1}^{\alpha} \mathrm{ben}(\mathcal{X}_i) \geq \sum_{i=1}^{k_-} \mathrm{ben}(\mathcal{X}_i) \geq \gamma_- + \gamma_+ - 1$.

Proof: (i) If $l_{\mathsf{w}}(\mathcal{X}) = 0$ then, by Eq. (4.5), the total mass of all the points in $\mathcal{X}$ is $\mathrm{mass}(\mathcal{X}) = h_{\mathsf{w}}(\mathcal{X}) + \xi \cdot l_{\mathsf{w}}(\mathcal{X}) = k_+ + 0 = k_+$, by Observation 4.4.1 (ii). Otherwise, we have $l_{\mathsf{w}}(\mathcal{X}) > 0$, and as such,

$$\mathrm{mass}(\mathcal{X}) = h_{\mathsf{w}}(\mathcal{X}) + \xi \cdot l_{\mathsf{w}}(\mathcal{X}) = h_{\mathsf{w}}(\mathcal{X}) + \frac{k_+ - h_{\mathsf{w}}(\mathcal{X}) - 1}{l_{\mathsf{w}}(\mathcal{X})} \cdot l_{\mathsf{w}}(\mathcal{X}) = k_+ - 1.$$

(ii) We have $\mathrm{mass}(\mathcal{X}) \geq k_+ - 1$, by (i), and $k_+ - k_- = \gamma_- + \gamma_+$, by definition. As such,

$$\sum_{i=1}^{k_-} \mathrm{ben}(\mathcal{X}_i) = \sum_{i=1}^{k_-} (\mathrm{mass}(\mathcal{X}_i) - 1) = \mathrm{mass}(\mathcal{X}) - k_- \geq k_+ - 1 - k_- = \gamma_- + \gamma_+ - 1.$$

Furthermore, since $\mathrm{ben}(\mathcal{X}_i) \le 0$, for $i = \alpha+1, \ldots, k_-$, we have

$$\sum_{i=1}^{\alpha} \mathrm{ben}(\mathcal{X}_i) \ge \sum_{i=1}^{\alpha} \mathrm{ben}(\mathcal{X}_i) + \sum_{i=\alpha+1}^{k_-} \mathrm{ben}(\mathcal{X}_i) = \sum_{i=1}^{k_-} \mathrm{ben}(\mathcal{X}_i)\,.$$

■

Claim 4.4.3 *(i) There exists $k' \le \alpha$ such that* $\sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) < \gamma_+ \le \sum_{t=1}^{k'} \mathrm{ben}(\mathcal{X}_t)$.

(ii) Step (i) of ClusterSparse *succeeds in computing $\mathcal{Y}_{k'}$ such that Eq. (4.2)$_{p36}$ holds.*

Proof: (i) By assumption, we have $\gamma_- \ge 3$, and as such, $\gamma_+ \le \gamma_- + \gamma_+ - 1 \le \sum_{t=1}^{\alpha} \mathrm{ben}(\mathcal{X}_t)$, by Claim 4.4.2 (ii). Therefore, k' is the first index for which this sum exceeds γ_+.

(ii) In step (i) of ClusterSparse, adding each point to $\mathcal{Y}_{k'}$ can increase the benefit of $\mathcal{Y}_{k'}$ by at most 1. This implies, by (i), that at some point, $\mathrm{BEN}(\overline{\mathcal{Y}}) = \sum_{i=1}^{k'-1} \mathrm{ben}(\mathcal{X}_i) + \mathrm{ben}(\mathcal{Y}_{k'})$ will fall inside the interval $[\gamma_+, \gamma_+ + 1)$, since $\mathcal{Y}_{k'} \subseteq \mathcal{X}_{k'}$. ■

Claim 4.4.4 *At least $k - k'$ heavy points are not included in $\mathcal{Y}$. Thus, in step (ii) of* ClusterSparse, *there are enough heavy points to be included in J, namely, $h_{\mathsf{w}}(J^{\mathsf{w}}) = k - k'$.*

Proof: Since, by definition, $\mathrm{mass}(B) = \mathrm{ben}(B) + 1$, for $B \subseteq \mathcal{X}$, we have

$$\mathrm{mass}(\mathcal{Y}) = \sum_{i=1}^{k'-1} \mathrm{mass}(\mathcal{X}_i) + \mathrm{mass}(\mathcal{Y}_{k'}) = \sum_{i=1}^{k'-1} \mathrm{ben}(\mathcal{X}_i) + \mathrm{ben}(\mathcal{Y}_{k'}) + k' = \mathrm{BEN}(\overline{\mathcal{Y}}) + k'.$$

Now, by Eq. (4.2)$_{\mathrm{p36}}$, which holds by Claim 4.4.3 (ii), this implies

$$\gamma_+ + k' \le \mathrm{mass}(\mathcal{Y}) < \gamma_+ + 1 + k'. \tag{4.6}$$

Since the mass of (all the copies) of a heavy point is one, it follows that the number of heavy points in $\mathcal{Y}$ is strictly smaller than $\gamma_+ + 1 + k'$ (or equivalently, it is at most $\gamma_+ + k'$). Now, since the total number of heavy points is $|\mathsf{C}_+| = k_+$, it follows that at least $k_+ - (\gamma_+ + k') = k - k'$ heavy points are not included in $\mathcal{Y}$, as the set $\mathcal{Y}$ does not partly-include any heavy point. ■

4.4.2 Bounding $\mathrm{cost}(\mathcal{Y})$

In this section, we prove that $\mathrm{cost}(\mathcal{Y}) = \nu(\mathsf{C}_-, \mathcal{Y}) = O(\mathrm{cost}(\mathcal{X}))$. The following technical lemma holds, since for any four real numbers $x, y \ge 0$ and $u, v > 0$ satisfying $\frac{x}{u} \le \frac{y}{v}$, we have $\frac{x}{u} \le \frac{x+y}{u+v} \le \frac{y}{v}$.

Lemma 4.4.5 *Given $x_1, \ldots, x_c \ge 0$ and $y_1, \ldots, y_c > 0$ such that $x_1/y_1 \le \ldots \le x_c/y_c$, we have that for any $1 \le b \le c$ and $0 < \beta \le 1$, it holds*

$$\frac{\sum_{t=1}^{b-1} x_t + \beta x_b}{\sum_{t=1}^{b-1} y_t + \beta y_b} \le \frac{\sum_{t=1}^{c} x_t}{\sum_{t=1}^{c} y_t}.$$

Claim 4.4.6 *We have that* $\mathrm{cost}(\mathcal{Y}_{k'}) \le \beta\, \mathrm{cost}(\mathcal{X}_{k'})$*, where* $\beta = \dfrac{\mathrm{mass}(\mathcal{Y}_{k'})}{\mathrm{mass}(\mathcal{X}_{k'})}$.

Proof: Observe that $0 < \beta \le 1$. Suppose that the set $\mathcal{X}_{k'}$ consists of u distinct points, $p_1, \ldots, p_u$, and furthermore, $\dfrac{\mathrm{cost}(p_i)}{\mathrm{mass}(p_i)} \le \dfrac{\mathrm{cost}(p_{i+1})}{\mathrm{mass}(p_{i+1})}$, for $i = 1, \ldots, u-1$. As such, $\mathcal{Y}_{k'}$ consists of $p_1, \ldots, p_{u'}$, for some $u' \le u$. By Lemma 4.4.5, we have

$$\frac{\mathrm{cost}(\mathcal{Y}_{k'})}{\mathrm{mass}(\mathcal{Y}_{k'})} = \frac{\sum_{i=1}^{u'} \mathsf{w}(p_i)\cdot \mathrm{cost}(p_i)}{\sum_{i=1}^{u'} \mathsf{w}(p_i)\cdot \mathrm{mass}(p_i)} \le \frac{\sum_{i=1}^{u} \mathsf{w}(p_i)\cdot \mathrm{cost}(p_i)}{\sum_{i=1}^{u} \mathsf{w}(p_i)\cdot \mathrm{mass}(p_i)} = \frac{\mathrm{cost}(\mathcal{X}_{k'})}{\mathrm{mass}(\mathcal{X}_{k'})},$$

implying that $\mathrm{cost}(\mathcal{Y}_{k'}) \le \dfrac{\mathrm{mass}(\mathcal{Y}_{k'})}{\mathrm{mass}(\mathcal{X}_{k'})}\mathrm{cost}(\mathcal{X}_{k'}) = \beta\mathrm{cost}(\mathcal{X}_{k'})$. ■

Lemma 4.4.7 *We have that* $\mathrm{cost}(\mathcal{Y}) \le 3\dfrac{\gamma_+}{\gamma_- + \gamma_+}\mathrm{cost}(\mathcal{X}) \le 36\mathsf{opt}$.

Proof: Let $\Delta = \sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) + \beta\, \mathrm{ben}(\mathcal{X}_{k'})$ and $\Gamma = \sum_{t=1}^{k'-1} \mathrm{cost}(\mathcal{X}_t) + \beta\, \mathrm{cost}(\mathcal{X}_{k'})$, where $\beta = \dfrac{\mathrm{mass}(\mathcal{Y}_{k'})}{\mathrm{mass}(\mathcal{X}_{k'})}$. We have

$$\begin{aligned}\beta\, \mathrm{ben}(\mathcal{X}_{k'}) &= \beta(\mathrm{mass}(\mathcal{X}_{k'}) - 1) = \mathrm{mass}(\mathcal{Y}_{k'}) - \beta = \Big(\mathrm{mass}(\mathcal{Y}_{k'}) - 1\Big) + (1-\beta)\\ &= \mathrm{ben}(\mathcal{Y}_{k'}) + 1 - \beta \le \mathrm{ben}(\mathcal{Y}_{k'}) + 1.\end{aligned}$$

Therefore,

$$\begin{aligned}\Delta &= \sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) + \beta\, \mathrm{ben}(\mathcal{X}_{k'}) \le \sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) + \mathrm{ben}(\mathcal{Y}_{k'}) + 1\\ &= \mathrm{BEN}(\overline{\mathcal{Y}}) + 1 < \gamma_+ + 2, \end{aligned} \tag{4.7}$$

by the construction of $\overline{\mathcal{Y}}$, see Eq. $(4.2)_{\mathrm{p36}}$. Since $\dfrac{\mathrm{cost}(\mathcal{X}_1)}{\mathrm{ben}(\mathcal{X}_1)} \le \ldots \le \dfrac{\mathrm{cost}(\mathcal{X}_\alpha)}{\mathrm{ben}(\mathcal{X}_\alpha)}$ and $1 \le k' \le \alpha$, we have, by Lemma 4.4.5, that

$$\frac{\Gamma}{\Delta} = \frac{\sum_{t=1}^{k'-1} \mathrm{cost}(\mathcal{X}_t) + \beta\, \mathrm{cost}(\mathcal{X}_{k'})}{\sum_{t=1}^{k'-1} \mathrm{ben}(\mathcal{X}_t) + \beta\, \mathrm{ben}(\mathcal{X}_{k'})} \le \frac{\sum_{t=1}^{\alpha} \mathrm{cost}(\mathcal{X}_t)}{\sum_{t=1}^{\alpha} \mathrm{ben}(\mathcal{X}_t)} \le \frac{\mathrm{cost}(\mathcal{X})}{\gamma_- + \gamma_+ - 1},$$

since $\sum_{t=1}^{\alpha} \mathrm{ben}(\mathcal{X}_t) \ge \gamma_- + \gamma_+ - 1$, by Claim 4.4.2 (ii), and $\sum_{t=1}^{\alpha} \mathrm{cost}(\mathcal{X}_t) \le \mathrm{cost}(\mathcal{X})$. This implies that

$$\Gamma \le \frac{\mathrm{cost}(\mathcal{X})}{\gamma_- + \gamma_+ - 1}\Delta < \frac{\gamma_+ + 2}{\gamma_- + \gamma_+ - 1}\mathrm{cost}(\mathcal{X})\,,$$

since $\Delta < \gamma_+ + 2$, see Eq. $(4.7)_{\mathrm{p43}}$. By Claim 4.4.6, $\mathrm{cost}(\mathcal{Y}_{k'}) \le \beta\, \mathrm{cost}(\mathcal{X}_{k'})$, and as such,

$$\begin{aligned}\mathrm{cost}(\mathcal{Y}) &= \sum_{t=1}^{k'-1} \mathrm{cost}(\mathcal{X}_t) + \mathrm{cost}(\mathcal{Y}_{k'}) \le \sum_{t=1}^{k'-1} \mathrm{cost}(\mathcal{X}_t) + \beta\, \mathrm{cost}(\mathcal{X}_{k'}) = \Gamma\\ &\le \frac{\gamma_+ + 2}{\gamma_- + \gamma_+ - 1}\mathrm{cost}(\mathcal{X}) \le 3\frac{\gamma_+}{\gamma_- + \gamma_+}\mathrm{cost}(\mathcal{X})\,,\end{aligned}$$

since $\dfrac{\gamma_+ + 2}{\gamma_- + \gamma_+ - 1} \le \dfrac{\gamma_+ + 3}{\gamma_- + \gamma_+} \le 3\dfrac{\gamma_+}{\gamma_- + \gamma_+}$ (implied by $\gamma_+ \ge 2$). Now, since $|\mathcal{X}|_{\mathsf{w}} \le n - m$, by the construction of $\mathcal{X}$, it holds that

$$\mathrm{cost}(\mathcal{X}) \le \mathsf{A}_m(\mathsf{C}_-, \mathsf{P}^{\mathsf{w}}) \le \left(3 + 9\frac{\gamma_- + \gamma_+}{\gamma_+}\right)\mathsf{opt},$$

by Corollary 4.3.7. Putting above two inequalities together, we obtain

$$\mathrm{cost}(\mathcal{Y}) \le 3\frac{\gamma_+}{\gamma_- + \gamma_+}\left(3 + 9\frac{\gamma_- + \gamma_+}{\gamma_+}\right)\mathsf{opt} \le 36\mathsf{opt}. \qquad \blacksquare$$

4.4.3 Proof of Lemma 4.4.14

Recall that the set J consists of the (distinct) $k - k'$ heaviest points excluded by $\mathcal{Y}$, and $Z = \mathcal{Y} \cup J^{\mathsf{w}}$ is the set of points clustered by the solution C output by ClusterSparse.

Claim 4.4.8 *If $\mathcal{X}$ does not contain any light point (namely, $l_{\mathsf{w}}(\mathcal{X}) = 0$), then $|Z|_{\mathsf{w}} \ge n - m$.*

Proof: Since $l_{\mathsf{w}}(\mathcal{X}) = 0$ and $\mathcal{Y} \subseteq \mathcal{X}$, it follows that $l_{\mathsf{w}}(\mathcal{Y}) = 0$. As such, by Eq. $(4.5)_{\mathrm{p}41}$, we have $\mathrm{mass}(\mathcal{Y}) = h_{\mathsf{w}}(\mathcal{Y}) + \xi \cdot l_{\mathsf{w}}(\mathcal{Y}) = h_{\mathsf{w}}(\mathcal{Y})$. On the other hand, by Eq. $(4.6)_{\mathrm{p}42}$, we have $\mathrm{mass}(\mathcal{Y}) \ge \gamma_+ + k'$, implying $h_{\mathsf{w}}(\mathcal{Y}) \ge \gamma_+ + k'$. Now, by the way ClusterSparse works, the set Z contains $h_{\mathsf{w}}(\mathcal{Y}) + h_{\mathsf{w}}(J^{\mathsf{w}}) \ge (\gamma_+ + k') + (k - k') = \gamma_+ + k = k_+$ heavy points, since $Z = \mathcal{Y} \cup J^{\mathsf{w}}$ and $h_{\mathsf{w}}(J^{\mathsf{w}}) = k - k'$, by Claim 4.4.4. That is, Z contains all the heavy points of C_+, which implies that $|Z|_{\mathsf{w}} \ge \mathsf{w}(\mathsf{C}_+) = n - m$. $\blacksquare$

As such, in the following, we assume that $l_{\mathsf{w}}(\mathcal{X}) > 0$. Recall that $h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)$ is the number of distinct heavy points in $\mathsf{P}^{\mathsf{w}} \setminus Z$. (Note that $\mathsf{P}^{\mathsf{w}} \setminus Z$ is the set of outliers for the clustering of Z computed by ClusterSparse.)

Lemma 4.4.9 *If $\mathcal{X}$ contains a light point, then $l_{\mathsf{w}}(\mathcal{Y}) \ge h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)/\xi$ (see Eq. $(4.1)_{p34}$).*

Proof: Since $Z = \mathcal{Y} \cup J^{\mathsf{w}}$ and $h_{\mathsf{w}}(J^{\mathsf{w}}) = k - k'$, by Claim 4.4.4, we have

$$\begin{aligned} h_{\mathsf{w}}(\mathcal{Y}) &= h_{\mathsf{w}}(Z) - h_{\mathsf{w}}(J^{\mathsf{w}}) = h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}}) - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) - (k - k') \\ &= k_+ - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) - k + k' = \gamma_+ + k' - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z), \end{aligned} \tag{4.8}$$

since $h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}}) = k_+$ and $k_+ - k = \gamma_+$. It follows that

$$\Big(\gamma_+ + k' - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)\Big) + \xi \cdot l_{\mathsf{w}}(\mathcal{Y}) = h_{\mathsf{w}}(\mathcal{Y}) + \xi \cdot l_{\mathsf{w}}(\mathcal{Y}) = \mathrm{mass}(\mathcal{Y}) \ge \gamma_+ + k',$$

by Eq. $(4.5)_{\mathrm{p}41}$ and Eq. $(4.6)_{\mathrm{p}42}$. This implies that $l_{\mathsf{w}}(\mathcal{Y}) \ge h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)/\xi$. $\blacksquare$

Claim 4.4.10 *If $\mathcal{X}$ contains a light point, then $h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) \le h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) - 1$.*

Proof: We have

$$\begin{aligned} \mathrm{mass}(\mathcal{X}) - \mathrm{mass}(\mathcal{Y}) &= \Big(h_{\mathsf{w}}(\mathcal{X}) + \xi \cdot l_{\mathsf{w}}(\mathcal{X})\Big) - \Big(h_{\mathsf{w}}(\mathcal{Y}) + \xi \cdot l_{\mathsf{w}}(\mathcal{Y})\Big) \\ &= h_{\mathsf{w}}(\mathcal{X}) - h_{\mathsf{w}}(\mathcal{Y}) + \xi\Big(l_{\mathsf{w}}(\mathcal{X}) - l_{\mathsf{w}}(\mathcal{Y})\Big) \\ &\ge h_{\mathsf{w}}(\mathcal{X}) - h_{\mathsf{w}}(\mathcal{Y}), \end{aligned}$$

since $l_{\mathsf{w}}(\mathcal{X}) \geq l_{\mathsf{w}}(\mathcal{Y})$ (implied by $\mathcal{X} \supseteq \mathcal{Y}$) and $\xi \geq 0$, by Observation 4.4.1 (iii). As such,

$$\begin{aligned} h_{\mathsf{w}}(\mathcal{X}) &\leq \text{mass}(\mathcal{X}) - \text{mass}(\mathcal{Y}) + h_{\mathsf{w}}(\mathcal{Y}) \\ &\leq (k_+ - 1) - (\gamma_+ + k') + (\gamma_+ + k' - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)) \\ &= k_+ - h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) - 1, \end{aligned}$$

since $\text{mass}(\mathcal{X}) = k_+ - 1$ (by Claim 4.4.2), $\text{mass}(\mathcal{Y}) \geq \gamma_+ + k'$ (by Eq. $(4.6)_{\text{p42}}$), and Eq. (4.8). It follows that

$$h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) \leq k_+ - h_{\mathsf{w}}(\mathcal{X}) - 1 = h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}}) - h_{\mathsf{w}}(\mathcal{X}) - 1 = h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) - 1.$$

since $h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}}) = k_+$. ■

Given a set $Q \subseteq \mathsf{P}^{\mathsf{w}}$ of heavy points, the *average weight* of Q is $|Q|_{\mathsf{w}} / h_{\mathsf{w}}(Q)$.

Observation 4.4.11 *Let Q and Q' be two sets of heavy points of P^{w}, where $Q \subseteq Q'$. Let S be a subset of Q', consisting of the $h_{\mathsf{w}}(S)$ lightest points in Q'. If $h_{\mathsf{w}}(S) \leq h_{\mathsf{w}}(Q)$ then*

$$\frac{|S|_{\mathsf{w}}}{h_{\mathsf{w}}(S)} \leq \frac{|Q|_{\mathsf{w}}}{h_{\mathsf{w}}(Q)}.$$

Given a set $Q \subseteq \mathsf{P}^{\mathsf{w}}$, let $H_w(Q)$ be the multiset of all the heavy points in Q, and $L_w(Q)$ be the set of all the light points in Q.

Lemma 4.4.12 *If $\mathcal{X}$ contains a light point, then $|H_w(\mathsf{P}^{\mathsf{w}} \setminus Z)|_{\mathsf{w}} \leq h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)/\xi$.*

Proof: By the construction of $\mathcal{X}$, there exists a point $p \in \mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}$ such that $|\mathcal{X}|_{\mathsf{w}} + \mathsf{w}(p) > n - m$. Let q be the heaviest point in $\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}$. Clearly, $\mathsf{w}(q) \geq \mathsf{w}(p)$, and as such,

$$|H_w(\mathcal{X})|_{\mathsf{w}} + |L_w(\mathcal{X})|_{\mathsf{w}} + \mathsf{w}(q) = |\mathcal{X}|_{\mathsf{w}} + \mathsf{w}(q) \geq |\mathcal{X}|_{\mathsf{w}} + \mathsf{w}(p) > n - m.$$

On the other hand, we have $|H_w(\mathcal{X})|_{\mathsf{w}} + |H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X})|_{\mathsf{w}} = |H_w(\mathsf{P}^{\mathsf{w}})|_{\mathsf{w}} = \mathsf{w}(\mathsf{C}_+) = n - m$. It thus follows that $|L_w(\mathcal{X})|_{\mathsf{w}} + \mathsf{w}(q) > n - m - |H_w(\mathcal{X})|_{\mathsf{w}} = |H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X})|_{\mathsf{w}}$, or equivalently,

$$|H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X})|_{\mathsf{w}} - \mathsf{w}(q) < |L_w(\mathcal{X})|_{\mathsf{w}}.$$

Let Q be the set of all the heavy points in $\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}$ except for q. As such, we have $|Q|_{\mathsf{w}} = |H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X})|_{\mathsf{w}} - \mathsf{w}(q) < |L_w(\mathcal{X})|_{\mathsf{w}} = l_{\mathsf{w}}(\mathcal{X})$ and $h_{\mathsf{w}}(Q) = h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) - 1$. Therefore, the average weight of Q is

$$\frac{|Q|_{\mathsf{w}}}{h_{\mathsf{w}}(Q)} < \frac{l_{\mathsf{w}}(\mathcal{X})}{h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) - 1} = \frac{l_{\mathsf{w}}(\mathcal{X})}{h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}}) - h_{\mathsf{w}}(\mathcal{X}) - 1} = \frac{l_{\mathsf{w}}(\mathcal{X})}{k_+ - h_{\mathsf{w}}(\mathcal{X}) - 1} = \frac{1}{\xi},$$

see Eq. $(4.1)_{\text{p34}}$. Note that $Q \subseteq H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) \subseteq H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{Y})$, since $\mathcal{Y} \subseteq \mathcal{X}$. By the way ClusterSparse works, $H_w(\mathsf{P}^{\mathsf{w}} \setminus Z)$ is a subset of $H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{Y})$, consisting of the $h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)$ lightest points in $H_w(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{Y})$. Furthermore, we have

$$h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z) \leq h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus \mathcal{X}) - 1 = h_{\mathsf{w}}(Q),$$

by Claim 4.4.10. Therefore, by Observation 4.4.11, we have

$$\frac{|H_w(\mathsf{P}^{\mathsf{w}} \setminus Z)|_{\mathsf{w}}}{h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)} \leq \frac{|Q|_{\mathsf{w}}}{h_{\mathsf{w}}(Q)} < \frac{1}{\xi}. \qquad \blacksquare$$

Lemma 4.4.14 (*restatement*) $|Z|_{\mathsf{w}} \geq n - m$.

Proof: Claim 4.4.8 handles the case $l_{\mathsf{w}}(\mathcal{X}) = 0$. So, consider the case when $l_{\mathsf{w}}(\mathcal{X}) > 0$. The total weight of Z is the number of light points in $\mathcal{Y}$ (note that there is no light points in J^{w}) plus the total weight of the heavy points in Z, namely,

$$\begin{aligned} |Z|_{\mathsf{w}} &= l_{\mathsf{w}}(Z) + |H_w(Z)|_{\mathsf{w}} = l_{\mathsf{w}}(\mathcal{Y}) + |H_w(Z)|_{\mathsf{w}} = l_{\mathsf{w}}(\mathcal{Y}) + |H_w(\mathsf{P}^{\mathsf{w}})|_{\mathsf{w}} - |H_w(\mathsf{P}^{\mathsf{w}} \setminus Z)|_{\mathsf{w}} \\ &\geq \frac{h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)}{\xi} + \mathsf{w}(\mathsf{C}_+) - \frac{h_{\mathsf{w}}(\mathsf{P}^{\mathsf{w}} \setminus Z)}{\xi} = n - m, \end{aligned}$$

by Lemma 4.4.9 and Lemma 4.4.12. ■

4.4.4 Putting things together

Lemma 4.4.13 *We have that* $\nu(\mathsf{C}, Z) \leq 36\mathsf{opt}$.

Proof: Since $Z = \mathcal{Y} \cup J^{\mathsf{w}}$ and $\mathsf{C} = \{f_1, \ldots, f_{k'}\} \cup J$, we have, by Lemma 4.4.7, that

$$\nu(\mathsf{C}, Z) \leq \nu\Big(\{f_1, \ldots, f_{k'}\}, \mathcal{Y}\Big) + \nu(J, J^{\mathsf{w}}) = \mathrm{cost}(\mathcal{Y}) + 0 \leq 36\mathsf{opt},$$

as $\mathcal{Y} \subseteq \bigcup_{i=1}^{k'} \mathcal{X}_i$, and f_i is the (nearest) facility of $\mathcal{X}_i$ in C_-. ■

The proof of the following lemma can be found in Appendix 4.4.3.

Lemma 4.4.14 *We have* $|Z|_{\mathsf{w}} \geq n - m$.

Lemma 4.4.15 *If* $\gamma_+ \geq 2$*, then one can compute, in* $O(n^2 \log^3 n)$ *time, a set* C *of* k *facilities such that* $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) \leq 39\mathsf{opt}$.

Proof: The algorithm is ClusterSparse, presented in Section 4.2.1. By Lemma 4.4.14, it holds $|Z|_{\mathsf{w}} \geq n - m$. Since $Z \subseteq \mathsf{P}^{\mathsf{w}}$, by Observation 4.3.1, we have $\mathsf{A}_m(\mathsf{C}, \mathsf{P}^{\mathsf{w}}) \leq \nu(\mathsf{C}, Z)$, which is at most $36\mathsf{opt}$, by Lemma 4.4.13. Now, Lemma 4.3.6 (i) implies that $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) \leq 3\mathsf{opt} + \mathsf{A}_m(\mathsf{C}, \mathsf{P}^{\mathsf{w}}) \leq 39\mathsf{opt}$. The overall running time is dominated by computing C_- and C_+, which takes $O(n^2 \log^3 n)$ time [CKMN01].
■

4.5 Correctness of ClusterDense ($\gamma_+ = 1$)

In this section, we show that, for the case $\gamma_+ = 1$, ClusterDense computes a solution C such that $|\mathsf{C}| = k$ and C is the desired approximation.

Definition 4.5.1 (Acceptable solution.) A facility set C of size k is an *acceptable solution* if $\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) \leq \mathsf{b}'\mathsf{opt}^{\mathsf{w}}$, where b' is an appropriate fixed constant.

We shall prove that $\mathsf{C} = \textsc{ClusterDense}(k, \mathsf{P}^{\mathsf{w}}, m)$ is an acceptable solution, which implies, by Lemma 4.3.6 (ii), that $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) = O\left(\mathsf{opt}\right)$. We remind the reader that in the penalty k-median with outliers problem (PMO), we are allowed to have more than m outliers, but every such additional outlier incurs an additional penalty ϱ.

Observation 4.5.2 *Let P be a set of n points, $C \subseteq P$, and $\varrho > 0$ be a penalty parameter.*

(i) $\mathcal{A}_m(C, P, \varrho) \leq \nu(C, M) + \varrho(n - m - |M|_{\mathsf{w}})$, for any $M \subseteq P$ such that $|M|_{\mathsf{w}} \leq n - m$.

(i) $\mathcal{A}_m(C, P, \varrho) \leq \nu(C, M)$, for any $M \subseteq P$ such that $|M|_{\mathsf{w}} \geq n - m$.

(iii) $\mathrm{opt}_{\mathsf{pmo}}(k, P, \varrho, m) \leq \mathrm{opt}_{\mathsf{mo}}(k, P, m)$.

4.5.1 The analysis of ClusterDense

$\varrho_0 = d_{min}/10$	$\varrho_i = 3^i \varrho_0$
$\Theta_i = \Theta(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$	$\overline{\Theta}_i = \Theta(\overline{B}_i, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$
$\Delta_i = n - m - \lvert\Theta_i\rvert$	$\overline{\Delta}_i = n - m - \left\lvert\overline{\Theta}_i\right\rvert_{\mathsf{w}}$
$\eta_i = \nu(B_i, \Theta_i)$	$\overline{\eta}_i = \nu\left(\overline{B}_i, \overline{\Theta}_i\right)$
$\omega_i = \mathcal{A}_m(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i)$	$\overline{\omega}_i = \mathcal{A}_m(\overline{B}_i, \mathsf{P}^{\mathsf{w}}, \varrho_i)$ $= \mathrm{opt}_{\mathsf{pmo}}(k, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$

Consider the algorithm $\textsc{LocalSearch}$ depicted in Figure 4.2. In the ith iteration, the facility set B_i is computed for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$. Let $\overline{B}_i$ be the *optimal* solution for the same instance. The notations used in this section are summarized in the table on the right. Here, $\Theta_i = \Theta(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$ denotes the set of points of $\mathbf{N}_{n-m}(B_i, \mathsf{P}^{\mathsf{w}})$ in distance strictly smaller than ϱ_i from B_i, namely, these are the points that contribute their true distances (from B_i) to $\mathcal{A}_m(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_i)$ (note that a point in $\mathbf{N}_{n-m}(C, P) \setminus \Theta_i$ pays only the penalty, as its distance to B_i is larger than ϱ_i). As such, Δ_i is the number of points that pay the penalty in the PMO clustering induced by B_i. By definition, we have

$$\omega_i = \nu(B_i, \Theta_i) + (n - m - |B_i|_{\mathsf{w}})\varrho_i = \eta_i + \Delta_i\, \varrho_i$$

and

$$\overline{\omega}_i = \nu\left(\overline{B}_i, \overline{\Theta}_i\right) + \left(n - m - \left|\overline{B}_i\right|_{\mathsf{w}}\right) \varrho_i = \overline{\eta}_i + \overline{\Delta}_i\, \varrho_i = \mathrm{opt}_{\mathsf{pmo}}(k, \mathsf{P}^{\mathsf{w}}, \varrho_i, m),$$

as $\overline{B}_i$ is the optimal solution.

The quantity $\overline{\Delta}$ is "dual" to the penalty parameter ϱ. In particular, $\overline{\Delta}$ is monotone decreasing as a function of ϱ[1].

Claim 4.5.3 *It holds that $\omega_0 = \overline{\omega}_0$, $\omega_1 = 3\overline{\omega}_0$, and $\omega_2 = 9\overline{\omega}_0$.*

Proof: It is easy to verify, by construction of P^{w}, that any k points of P^{w} have total weight at most $n - m$. As such, when $j = 0, 1, 2$, it holds that $\Theta(C, \mathsf{P}^{\mathsf{w}}, \varrho_j, m) = C^{\mathsf{w}}$, for any $C \subseteq \mathsf{P}^{\mathsf{w}}$ satisfying $|C| = k$, since $\varrho_j \leq 9\, d_{min}/10 < d_{min}$ (which implies that no point in $\mathsf{P}^{\mathsf{w}} \setminus C^{\mathsf{w}}$ is in distance smaller than ϱ_j to C). Therefore, when $j = 0, 1, 2$, we have

$$\mathcal{A}_m(C, \mathsf{P}^{\mathsf{w}}, \varrho_j) = \nu(C, C^{\mathsf{w}}) + (n - m - |C^{\mathsf{w}}|_{\mathsf{w}})\varrho_j = (n - m - |C^{\mathsf{w}}|_{\mathsf{w}})\varrho_j.$$

[1] We sketch the proof here for $\overline{\Delta}_{i+1} \leq \overline{\Delta}_i$. Indeed, by Observation 4.5.2 (i), it is not hard to verify that $\overline{\eta}_i + \overline{\Delta}_i\, \varrho_i \leq \overline{\eta}_{i+1} + \overline{\Delta}_{i+1}\, \varrho_i$ and $\overline{\eta}_{i+1} + \overline{\Delta}_{i+1}\, \varrho_{i+1} \leq \overline{\eta}_i + \overline{\Delta}_i\, \varrho_{i+1}$. Adding these two inequalities together, we obtain $\overline{\Delta}_{i+1}(\varrho_{i+1} - \varrho_i) \leq \overline{\Delta}_i(\varrho_{i+1} - \varrho_i)$. Since $\varrho_{i+1} - \varrho_i = 3\varrho_i - \varrho_i > 0$, this implies that $\overline{\Delta}_{i+1} \leq \overline{\Delta}_i$.

This implies that $B_0 = \overline{B}_0 = B_1 = B_2 = \mathsf{H}$, because H is the set of the k heaviest points. Now the claim follows, since $\varrho_2 = 9\varrho_0$ and $\varrho_1 = 3\varrho_0$. ■

Claim 4.5.4 *For $i \geq 0$, it holds that (i) $\omega_{i+1} - \omega_i \leq 2\Delta_i\, \varrho_i$ and (ii) $2\overline{\Delta}_{i+1}\, \varrho_i \leq \overline{\omega}_{i+1} - \overline{\omega}_i$.*

Proof: (i) We have $\omega_{i+1} = \mathcal{A}_m(B_{i+1}, \mathsf{P}^{\mathsf{w}}, \varrho_{i+1}) \leq \mathcal{A}_m(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_{i+1})$, since B_{i+1} is computed by a local search starting from B_i. In addition, by Observation 4.5.2 (i), we have

$$\mathcal{A}_m(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_{i+1}) \leq \nu(B_i, \Theta_i) + (n - m - |\Theta_i|_{\mathsf{w}})\varrho_{i+1} = \eta_i + \Delta_i\, \varrho_{i+1}.$$

It follows that

$$\omega_{i+1} \leq \eta_i + \Delta_i\, \varrho_{i+1} = \eta_i + 3\Delta_i\, \varrho_i = \omega_i + 2\Delta_i\, \varrho_i\ ,$$

since $\varrho_{i+1} = 3\varrho_i$ and $\omega_i = \eta_i + \Delta_i\, \varrho_i$.

(ii) We have $\overline{\omega}_i = \mathcal{A}_m(\overline{B}_i, \mathsf{P}^{\mathsf{w}}, \varrho_i) \leq \mathcal{A}_m(\overline{B}_{i+1}, \mathsf{P}^{\mathsf{w}}, \varrho_i)$, since $\overline{B}_i$ is the optimal solution for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho_i, m)$. By Observation 4.5.2 (i), we have

$$\mathcal{A}_m(B_i, \mathsf{P}^{\mathsf{w}}, \varrho_{i+1}) \leq \nu(B_i, \Theta_i) + (n - m - |\Theta_i|_{\mathsf{w}})\varrho_{i+1} = \eta_i + \Delta_i\, \varrho_{i+1}.$$

It follows that

$$\overline{\omega}_i \leq \overline{\eta}_{i+1} + \overline{\Delta}_{i+1}\, \varrho_i = (\overline{\eta}_{i+1} + 3\overline{\Delta}_{i+1}\, \varrho_i) - 2\overline{\Delta}_{i+1}\, \varrho_i = \overline{\omega}_{i+1} - 2\overline{\Delta}_{i+1}\, \varrho_i\ ,$$

since $\overline{\omega}_{i+1} = \overline{\eta}_{i+1} + \overline{\Delta}_{i+1}\, \varrho_{i+1} = \overline{\eta}_{i+1} + 3\overline{\Delta}_{i+1}\, \varrho_i$. ■

The proof of the following lemma can be found in Section 4.5.2.

Lemma 4.5.5 *If $\omega_i \leq 9\mathsf{opt}^{\mathsf{w}}$ and there is no acceptable solution in $\mathcal{H} \cup \mathtt{N}(B_i)$, then $\Delta_i \leq \overline{\Delta}_{i-1}$.*

Naturally, when the penalty parameter exceeds d_{max}, no point would pay the penalty in the solution computed by ClusterDense. As such, before $\varrho_i > 3d_{max}$, we would have $\Delta_i = 0$ and thus, ClusterDense terminates. Since $\varrho_0 = d_{min}/10$ and $d_{max}/d_{min} = O(n^2)$, this implies that it terminates after $O(\log n)$ calls to LocalSearch (with gradually increasing penalty parameters).

Lemma 4.5.6 *If there is no acceptable solution in $\mathcal{H} \cup \bigcup_{t=0}^{I} \mathtt{N}(B_t)$, then $\omega_j \leq 9\mathsf{opt}^{\mathsf{w}}$, for $j = 0, \ldots, I$, where I is the smallest index such that $\Delta_I = 0$.*

Proof: By induction on j. For the base cases $j = 0, 1$, and 2, Claim 4.5.3 implies that $\omega_j \leq 9\overline{\omega}_0 = 9\mathrm{opt}_{\mathsf{pmo}}(k, \mathsf{P}^{\mathsf{w}}, \varrho_0, m) \leq 9\mathsf{opt}^{\mathsf{w}}$, by Observation 4.5.2 (iii). Thus, assume that the claim holds when $0 \leq j \leq i-1$, where $3 \leq i \leq I$. We need to show that $\omega_i \leq 9\mathsf{opt}^{\mathsf{w}}$.

By Lemma 4.5.5, we have that $\Delta_t \leq \overline{\Delta}_{t-1}$, for $1 \leq t \leq i-1$, since $\omega_t \leq 9\mathsf{opt}^{\mathsf{w}}$ by the induction hypothesis. Therefore, since $\varrho_t = 9\varrho_{t-2}$, for $2 \leq t \leq i-1$, we have

$$\omega_{t+1} - \omega_t \leq 2\Delta_t\, \varrho_t \leq 2\overline{\Delta}_{t-1}\, \varrho_t = 18\overline{\Delta}_{t-1}\, \varrho_{t-2} \leq 9(\overline{\omega}_{t-1} - \overline{\omega}_{t-2})\,,$$

by Claim 4.5.4. Summing this inequality, for $t = 2, \ldots, i-1$, we obtain $\omega_i - \omega_2 \leq 9(\overline{\omega}_{i-2} - \overline{\omega}_0)$. This implies $\omega_i \leq 9(\overline{\omega}_{i-2} - \overline{\omega}_0) + \omega_2 = 9\overline{\omega}_{i-2} \leq 9\mathsf{opt}^{\mathsf{w}}$ since $\omega_2 = 9\overline{\omega}_0$, by Claim 4.5.3, and $\overline{\omega}_{i-2} = \mathrm{opt}_{\mathsf{pmo}}(k, \mathsf{P}^{\mathsf{w}}, \varrho_{i-2}, m) \leq \mathsf{opt}^{\mathsf{w}}$, by Observation 4.5.2 (iii). ■

Claim 4.5.7 *The set $\mathcal{H} \cup \bigcup_{t=0}^{I} \mathtt{N}(B_t)$ contains an acceptable solution, where I is the smallest index such that $\Delta_I = 0$.*

Proof: Assume for the sake of contradiction that $\mathcal{H} \cup \bigcup_{t=0}^{I} \mathtt{N}(B_t)$ does not contain an acceptable solution. Since $\Delta_I = 0$, it follows that $|\Theta_I|_{\mathsf{w}} = n - m - \Delta_I = n - m$ and $\omega_I = \nu(B_I, \Theta_I) + \varrho_I \Delta_I = \nu(B_I, \Theta_I)$. Therefore, by Observation 4.3.1, $\mathsf{A}_m(B_I, \mathsf{P}^{\mathsf{w}}) \leq \nu(B_I, \Theta_I) = \omega_I \leq 9\mathsf{opt}^{\mathsf{w}}$, by Lemma 4.5.6. However, by definition, this implies that B_I is an acceptable solution. A contradiction. ■

Lemma 4.5.8 *If $\gamma_+ = 1$, then one can compute, in $O(k^2(k+m)^2 n^3 \log n)$ time, a set C of k points such $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) \leq (4\mathsf{b}' + 3)\mathsf{opt}$, where b' is the constant in Definition 4.5.1.*

Proof: The algorithm is ClusterDense, described in Section 4.2.2. By Claim 4.5.7, we have $\mathsf{A}_m(\mathsf{C}, \mathsf{P}^{\mathsf{w}}) \leq \mathsf{b}'\mathsf{opt}^{\mathsf{w}}$, where C is the solution computed by ClusterDense. Now, Lemma 4.3.6 (ii) implies $\mathsf{A}_m(\mathsf{C}, \mathsf{P}) \leq (4\mathsf{b}' + 3)\mathsf{opt}$.

The overall running time of ClusterDense is dominated by the calls to LocalSearch. As discussed above, ClusterDense terminates after $O(\log n)$ calls of LocalSearch. There are $O(nd_{max}/(d_{min}/30)) = O(n^3)$ local search steps done by LocalSearch, because nd_{max} is an upper bound of the cost for any valid solution for $\mathsf{MO}(k, \mathsf{P}^{\mathsf{w}}, m)$ and $\varrho_0/3 = d_{min}/30$ is a lower bound on the improvement a local search step makes. Each local search step in LocalSearch needs to check $O(k(k+m))$ neighbors and each check (namely, to see if a neighbor facility set is better than the current solution) takes $O(k(k+m))$ time, since there are only $k_+ + m = O(k+m)$ distinct points in P^{w}, see Remark 4.3.5. Hence, the total running time is $O(k^2(k+m)^2 n^3 \log n)$. ■

4.5.2 Proof of Lemma 4.5.5

Notations and assumptions

$F = \text{LocalSearch}(B, \mathsf{P}^{\mathsf{w}}, 3\varrho)$, where B is an arbitrary set of k facilities.
$U = \Theta(F, \mathsf{P}^{\mathsf{w}}, 3\varrho, m)$.
$\Delta = n - m - \lvert U \rvert_{\mathsf{w}}$.
U_v: the points of U served by v, for $v \in F$.
$\overline{F}$: Optimal solution for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho, m)$.
$\overline{U} = \Theta(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho, m)$.
$\overline{\Delta} = n - m - \lvert \overline{U} \rvert_{\mathsf{w}}$.
$\overline{U}_{\overline{x}}$: the points of $\overline{U}$ served by $\overline{x}$, for $\overline{x} \in \overline{F}$.

Given a parameter $\varrho \geq 0$ and an arbitrary facility set B satisfying $|B| = k$, let $F = \text{LocalSearch}(B, \mathsf{P}^{\mathsf{w}}, 3\varrho)$. And let $\overline{F}$ be the globally optimal solution for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho, m)$. The notations used in this section are summarized in the table on the right.

In the remainder of this section, we prove that $\Delta \leq \overline{\Delta}$ under the following assumptions:

(A1): $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \leq 9\mathsf{opt}^{\mathsf{w}}$.

(A2): $\mathcal{H} \cup \mathtt{N}(F)$ does not contain an acceptable solution.

(A3): $\Delta > 0$, that is, $|U|_{\mathsf{w}} < n - m$. (If $\Delta = 0$, then the claim trivially holds, since $\overline{\Delta} \geq 0$.)

Specifically, the claim is that the LocalSearch solution (with penalty parameter 3ϱ) penalizes no more points than the optimal solution (with penalty parameter ϱ). In other words, the balls of radius 3ϱ centered at the facilities of the LocalSearch solution cover no less points than the balls of radius ϱ centered at the facilities of the optimal solution.

Proof of Lemma 4.5.5

Our proof is remotely similar to the approach used by Arya *et al.* [AGK$^+$04]. The idea is to establish a bijection $\pi : F \to \overline{F}$ such that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}}$ holds for all $v \in F$. The quantity $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}}$ quantifies by how much $\overline{U}$ would grow (in size) if the cluster U_v is added to $\overline{U}$, and $\left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}}$ quantifies by how much U would grow if $\overline{U}_{\pi(v)}$ is added to U. Therefore, $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}}$ implies, in some sense, that U_v is more "valuable" than $\overline{U}_{\pi(v)}$. In particular, if π has this property, then

$$|U|_{\mathsf{w}} - \left|\overline{U}\right|_{\mathsf{w}} = \left|U \setminus \overline{U}\right|_{\mathsf{w}} - \left|\overline{U} \setminus U\right|_{\mathsf{w}} = \sum_{v \in F} \left(\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} - \left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}} \right) \geq 0, \tag{4.9}$$

and thus, $-|U|_{\mathsf{w}} \leq -\left|\overline{U}\right|_{\mathsf{w}}$. This implies $\Delta = n - m - |U|_{\mathsf{w}} \leq n - m - \left|\overline{U}\right|_{\mathsf{w}} = \overline{\Delta}$, by definition.

Lemma 4.5.9 *Under the assumptions of Section 4.5.2, we have that*
(i) $\nu(F, U) \leq \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \leq 9\mathsf{opt}^{\mathsf{w}}$, *and*
(ii) $\nu\left(\overline{F}, \overline{U}\right) \leq \mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) \leq \mathsf{opt}^{\mathsf{w}}$.

Proof: (i) The first inequality holds because $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) = \nu(F, U) + 3\varrho\Delta$ and $\varrho, \Delta \geq 0$. The second inequality holds by assumption (A1).

(ii) The first inequality holds by the same argument as (i). As for the second inequality, since $\overline{F}$ is optimal for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho, m)$, we have $\mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) = \mathrm{opt}_{\mathsf{pmo}}(k, \mathsf{P}^{\mathsf{w}}, \varrho, m) \leq \mathsf{opt}^{\mathsf{w}}$, by Observation 4.5.2 (iii).
■

The proof of the following lemma can be found in Section 4.5.2.

Lemma 4.5.10 *Under the assumptions of Section 4.5.2, for any* $\overline{x}, \overline{y} \in \overline{F}$ *and* $q \in \mathsf{P}$, *we have* $\nu\left(q, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) \geq 15\mathsf{opt}^{\mathsf{w}}$.

Intuitively, Lemma 4.5.10 holds because $|\mathsf{C}_+| = k + 1$ and $\mathsf{w}(\mathsf{C}_+) = n - m$. Indeed, assume that such $\overline{x}, \overline{y}$, and q satisfying $\nu\left(q, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) = O(\mathsf{opt}^{\mathsf{w}})$ exists, namely, we can use one single facility (i.e., q) to serve $\overline{U}_{\overline{x}}$ and $\overline{U}_{\overline{y}}$ together "cheaply". It is not hard to argue that the size of $\overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}$ is larger than two heavy points, say h_1 and h_2. Since there are $k + 1$ heavy points in total, and their total weight is $n - m$, we can use the $k - 1$ heavy points (other than h_1 and h_2) as the $k - 1$ clusters. These $k - 1$ clusters together with q (which serves $\overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}$) would be an acceptable solution, contradicting assumption (A2).

Definition 4.5.11 (Match, capture, and prisoner.) Two facilities $v \in F$ and $\overline{x} \in \overline{F}$ *overlap* if $U_v \cap \overline{U}_{\overline{x}} \neq \emptyset$. We construct a graph $\mathcal{G} = (F \cup \overline{F}, \mathcal{E})$, where the edge $v\overline{x} \in \mathcal{E}$ if v and $\overline{x}$ overlap. The *degree* of $u \in F \cup \overline{F}$ is denoted by $\deg(u)$.

A facility $v \in F$ and a facility $\overline{x} \in \overline{F}$ *match*, if $v\overline{x} \in \mathcal{E}$ and $\deg(v) = \deg(\overline{x}) = 1$.

A facility $v \in F$ *captures* a facility $\overline{x} \in \overline{F}$, if v is the nearest neighbor to $\overline{x}$ in F and $\mathsf{d}(v, \overline{x}) < 2\varrho$. In this case, $\overline{x}$ is a *prisoner* of v.

Observation 4.5.12 *Under the assumptions of Section 4.5.2, we have $|U|_{\mathsf{w}} < n - m$, and as such, all the points of P^{w} in distance at most 3ϱ from F are in U.*

Claim 4.5.13 *Under the assumptions of Section 4.5.2, if v captures $\overline{x}$ then $v\overline{x} \in \mathcal{E}$.*

Proof: Since $\mathsf{d}(F, \overline{x}) \le \mathsf{d}(v, \overline{x}) < 2\varrho$, we have, by Observation 4.5.12, that $\overline{x} \in U$. Now, since the nearest neighbor to $\overline{x}$ in F is v, it follows that $\overline{x}$ is in the cluster of v, namely $\overline{x} \in U_v$. Therefore, we have $\overline{x} \in \overline{U}_{\overline{x}} \cap U_v$, which implies the claim. ■

Claim 4.5.14 *For $v \in F$ and $\overline{x} \in \overline{F}$, if $\overline{x}$ is a prisoner of v, then*
(i) $\overline{U}_{\overline{x}} \subseteq U$ (that is, $\left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}} = 0$), and
(ii) for any $p \in \overline{U}_{\overline{x}}$, it holds that $\mathsf{d}(v, p) \le \mathsf{d}(F, p) + 2\mathsf{d}(\overline{F}, p)$.

Proof: (i) For a point $p \in \overline{U}_{\overline{x}}$, it holds, by the triangle inequality, that $\mathsf{d}(F, p) \le \mathsf{d}(v, p) \le \mathsf{d}(v, \overline{x}) + \mathsf{d}(\overline{x}, p) \le 2\varrho + \varrho = 3\varrho$. Thus, by Observation 4.5.12, we have $p \in U$.

(ii) Fix a point $p \in \overline{U}_{\overline{x}}$, and let s be the nearest neighbor to p in F. Since v captures $\overline{x}$, it holds that the nearest neighbor to $\overline{x}$ in F is v, and as such $\mathsf{d}(v, \overline{x}) \le \mathsf{d}(s, \overline{x})$. Therefore, by the triangle inequality, we have

$$\begin{aligned}
\mathsf{d}(v, p) &\le \mathsf{d}(v, \overline{x}) + \mathsf{d}(\overline{x}, p) \le \mathsf{d}(s, \overline{x}) + \mathsf{d}(\overline{x}, p) \le (\mathsf{d}(s, p) + \mathsf{d}(p, \overline{x})) + \mathsf{d}(\overline{x}, p) \\
&= \mathsf{d}(F, p) + 2\mathsf{d}(\overline{x}, p) = \mathsf{d}(F, p) + 2\mathsf{d}(\overline{F}, p),
\end{aligned}$$

since $p \in \overline{U}_{\overline{x}}$. ■

Claim 4.5.15 *Under the assumptions of Section 4.5.2, any facility in $\overline{F}$ can be a prisoner of at most one facility in F, and any facility of F can capture at most one facility of $\overline{F}$.*

Proof: The first assertion follows from the definition, since a prisoner always belong to its nearest neighbor in F (which is distinct, as all distances are distinct). As for the second claim, let $v \in F$, and assume, for the sake of contradiction, that v captures two facilities $\overline{x}, \overline{y} \in \overline{F}$. By Claim 4.5.14 (ii), we have

$$\begin{aligned}
\nu\left(v, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) &= \sum_{p \in \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}} \mathsf{d}(v, p) \le \sum_{p \in \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}} (\mathsf{d}(F, p) + 2\mathsf{d}(\overline{F}, p)) \\
&\le \sum_{p \in U} \mathsf{d}(F, p) + \sum_{p \in \overline{U}} 2\mathsf{d}(\overline{F}, p) = \nu(F, U) + 2\nu\left(\overline{F}, \overline{U}\right),
\end{aligned}$$

since $\overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}} \subseteq \overline{U}$ and $\overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}} \subseteq U$, by Claim 4.5.14 (i). Now, By Lemma 4.5.9, we have

$$\nu\left(v, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) \le \nu(F, U) + 2\nu\left(\overline{F}, \overline{U}\right) \le 9\mathsf{opt}^{\mathsf{w}} + 2\mathsf{opt}^{\mathsf{w}} \le 11\mathsf{opt}^{\mathsf{w}},$$

contradicting Lemma 4.5.10. ■

Definition 4.5.16 Let $F_C \subseteq F$ be the set of facilities that capture some facilities of $\overline{F}$, and let $\overline{F}_C \subseteq \overline{F}$ be the corresponding set of prisoners. By Claim 4.5.15, there exists a bijection $\pi_C : F_C \to \overline{F}_C$ such that v captures $\pi_C(v)$ for each $v \in F_C$.

Let $F_M \subseteq F \setminus F_C$ be the set of facilities which match some facilities in $\overline{F} \setminus \overline{F}_C$, and let $\overline{F}_M \subseteq \overline{F} \setminus \overline{F}_C$ be the set of facilities which match some facilities in F_M. It follows from the definition that there exists a bijection $\pi_P : F_M \to \overline{F}_M$ such that v and $\pi_P(v)$ matches each other, for every $v \in F_M$.

Let $F_L = F \setminus (F_C \cup F_M)$ and $\overline{F}_L = \overline{F} \setminus (\overline{F}_C \cup \overline{F}_M)$. Let $\pi_L : F_L \to \overline{F}_L$ be an arbitrary bijection.

Let $\pi : F \to \overline{F}$ be the bijection formed together by π_C, π_P, and π_L. See Figure 4.3.

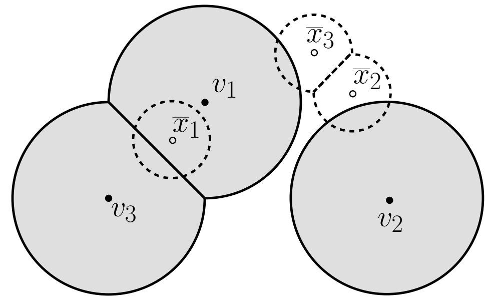


Figure 4.3: $F = \{v_1, v_2, v_3\}$ and $\overline{F} = \{\overline{x}_1, \overline{x}_2, \overline{x}_3\}$. The area inside the circles represents the points in U, the area inside the dashed circles represents the points in $\overline{U}$. Here, v_1 captures $\overline{x}_1$, and v_2 matches $\overline{x}_2$ but does not capture $\overline{x}_2$. We have $\pi(v_1) = \overline{x}_1$, $\pi(v_2) = \overline{x}_2$, and $\pi(v_3) = \overline{x}_3$.

We next establish that, for all $v \in F$, it holds $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}}$, which proves Eq. $(4.9)_{\text{p50}}$ and thus implies Lemma 4.5.5. In fact, since π_L is an arbitrary bijection (between F_L and $\overline{F}_L$), our proof would imply the stronger property that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$, for any $v \in F_L$ and $\overline{x} \in \overline{F}_L$. (However, our proof actually does not require this stronger property.)

The following lemma is implied immediately by Claim 4.5.14 (i).

Lemma 4.5.17 *If $v \in F_C$ and $\overline{x} = \pi(v)$ then $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}} = 0$.*

Lemma 4.5.18 *Under the assumptions of Section 4.5.2, there does not exist a multiset $M \subseteq \mathsf{P}^{\mathsf{w}}$ and a set $C \in \mathcal{H} \cup \mathtt{N}(F)$ such that $|M|_{\mathsf{w}} \geq n - m$ and $\nu(C, M) \leq \mathsf{b}'\mathsf{opt}^{\mathsf{w}}$, where b' is the constant in Definition 4.5.1.*

Proof: Assume for the sake of contradiction that such a set exists. Then, by Observation 4.3.1, it holds $\mathsf{A}_m(C, \mathsf{P}^{\mathsf{w}}) \leq \nu(C, M) \leq \mathsf{b}'\mathsf{opt}^{\mathsf{w}}$, which implies that C is an acceptable solution. This contradicts the assumption (A2) that $\mathcal{H} \cup \mathtt{N}(F)$ does not contain such a solution. ■

Let $U_{v\to\overline{x}} = (U \setminus U_v) \cup \overline{U}_{\overline{x}}$ and $\overline{U}_{\overline{x}\to v} = (\overline{U} \setminus \overline{U}_{\overline{x}}) \cup U_v$.

Lemma 4.5.19 *If $\left|\overline{U}_{\overline{x}\to v}\right|_{\mathsf{w}} - \left|\overline{U}\right|_{\mathsf{w}} \geq |U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}$, then $\left|\overline{U}_{\overline{x}\to v}\right|_{\mathsf{w}} \geq \left|\overline{U}\right|_{\mathsf{w}}$.*

Proof: Assume for the sake of contradiction that $\left|\overline{U}_{\overline{x}\to v}\right|_{\mathsf{w}} < \left|\overline{U}\right|_{\mathsf{w}}$. Let $F_{v\to\overline{x}} = F - v + \overline{x}$, where the notation $F - v + \overline{x}$ refers to $(F \setminus \{v\}) \cup \{\overline{x}\}$. We have

$$\begin{aligned}
\nu(F_{v\to\overline{x}}, U_{v\to\overline{x}}) - \nu(F, U) &\leq \Big(\nu(F - v, U \setminus U_v) + \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big)\Big) - \Big(\nu(F - v, U \setminus U_v) + \nu(v, U_v)\Big) \\
&= \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) - \nu(v, U_v)\,. \qquad (4.10)
\end{aligned}$$

This implies that

$$\begin{aligned}\nu(F_{v\to\overline{x}}, U_{v\to\overline{x}}) &\le \nu(F, U) - \nu(v, U_v) + \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) \le \nu(F, U) + \nu\big(\overline{F}, \overline{U}\big)\\ &\le 9\mathsf{opt}^{\mathsf{w}} + \mathsf{opt}^{\mathsf{w}} \le 10\mathsf{opt}^{\mathsf{w}}, \end{aligned} \tag{4.11}$$

by Lemma 4.5.9. If $|U_{v\to\overline{x}}|_{\mathsf{w}} \ge n - m$ then $F_{v\to\overline{x}}$ is an acceptable solution, contradicting Lemma 4.5.18. Thus, we have $|U_{v\to\overline{x}}|_{\mathsf{w}} < n - m$. Now, by Observation 4.5.2 (i), we have $\mathcal{A}_m(F_{v\to\overline{x}}, \mathsf{P}^{\mathsf{w}}, 3\varrho) \le \nu(F_{v\to\overline{x}}, U_{v\to\overline{x}}) + 3\varrho(n - m - |U_{v\to\overline{x}}|_{\mathsf{w}})$. Therefore,

$$\begin{aligned} D &= \mathcal{A}_m(F_{v\to\overline{x}}, \mathsf{P}^{\mathsf{w}}, 3\varrho) - \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho)\\ &\le \Big(\nu(F_{v\to\overline{x}}, U_{v\to\overline{x}}) + 3\varrho\cdot\big(n - m - |U_{v\to\overline{x}}|_{\mathsf{w}}\big)\Big) - \Big(\nu(F, U) + 3\varrho\cdot\big(n - m - |U|_{\mathsf{w}}\big)\Big)\\ &= \nu(F_{v\to\overline{x}}, U_{v\to\overline{x}}) - \nu(F, U) + 3\varrho\cdot\big(|U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}\big)\\ &\le \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) - \nu(v, U_v) + 3\varrho\cdot\big(|U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}\big)\,, \end{aligned}$$

by Eq. (4.10). Moreover, since F is the solution computed by LocalSearch, we have that $D \ge -(3\varrho)/3 = -\varrho$ and as such,

$$\nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) - \nu(v, U_v) + 3\varrho\cdot\big(|U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}\big) \ge D \ge -\varrho. \tag{4.12}$$

Let $\overline{F}_{\overline{x}\to v} = \overline{F} - \overline{x} + v$. Since $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} < \big|\overline{U}\big|_{\mathsf{w}} \le n - m$ (by assumption), arguing as above, we have

$$\overline{D} = \mathcal{A}_m(\overline{F}_{\overline{x}\to v}, \mathsf{P}^{\mathsf{w}}, \varrho) - \mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) \le \nu(v, U_v) - \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) + \varrho\cdot\big(\big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}}\big)\,.$$

Moreover, since $\overline{F}$ is the optimal solution for $\mathsf{PMO}(k, \mathsf{P}^{\mathsf{w}}, \varrho, m)$, we have $\overline{D} \ge 0$. It follows

$$\nu(v, U_v) - \nu\big(\overline{x}, \overline{U}_{\overline{x}}\big) + \varrho\cdot\big(\big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}}\big) \ge 0. \tag{4.13}$$

Now, adding Eq. (4.12) and Eq. (4.13) together, we obtain

$$3\varrho\cdot\big(|U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}\big) + \varrho\cdot\big(\big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}}\big) \ge -\varrho.$$

Since $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} - \big|\overline{U}\big|_{\mathsf{w}} \ge |U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}$, it follows that

$$3\varrho\cdot\big(\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} - \big|\overline{U}\big|_{\mathsf{w}}\big) + \varrho\cdot\big(\big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}}\big) \ge 3\varrho\cdot\big(|U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}\big) + \varrho\cdot\big(\big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}}\big) \ge -\varrho,$$

or equivalently, $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} - \big|\overline{U}\big|_{\mathsf{w}} \ge -1/2$. This implies that $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} \ge \big|\overline{U}\big|_{\mathsf{w}}$, contradicting our assumption $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} < \big|\overline{U}\big|_{\mathsf{w}}$. ■

Lemma 4.5.20 *Under the assumptions of Section 4.5.2, if $v \in F_M$ and $\overline{x} = \pi(v)$ then $\big|U_v \setminus \overline{U}\big|_{\mathsf{w}} \ge \big|\overline{U}_{\overline{x}} \setminus U\big|_{\mathsf{w}}$.*

Proof: Since v and $\overline{x}$ match each other, $\overline{x}$ is the only facility in $\overline{F}$ that overlaps with v, and as such, $|U_{v\to\overline{x}}|_{\mathsf{w}} = |U \setminus U_v|_{\mathsf{w}} + \big|\overline{U}_{\overline{x}}\big|_{\mathsf{w}} = |U|_{\mathsf{w}} - |U_v|_{\mathsf{w}} + \big|\overline{U}_{\overline{x}}\big|_{\mathsf{w}}$. Similarly, we have $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} = \big|\overline{U}\big|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}}\big|_{\mathsf{w}} + |U_v|_{\mathsf{w}}$. It thus follows that $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} - \big|\overline{U}\big|_{\mathsf{w}} = |U_v|_{\mathsf{w}} - \big|\overline{U}_{\overline{x}}\big|_{\mathsf{w}} = |U|_{\mathsf{w}} - |U_{v\to\overline{x}}|_{\mathsf{w}}$. Now, by Lemma 4.5.19, we have $\big|\overline{U}_{\overline{x}\to v}\big|_{\mathsf{w}} \ge \big|\overline{U}\big|_{\mathsf{w}}$, which implies $|U_v|_{\mathsf{w}} \ge \big|\overline{U}_{\overline{x}}\big|_{\mathsf{w}}$. Therefore, we have $\big|U_v \setminus \overline{U}\big|_{\mathsf{w}} = |U_v|_{\mathsf{w}} - \big|U_v \cap \overline{U}_{\overline{x}}\big|_{\mathsf{w}} \ge$

$\left|\overline{U}_{\overline{x}}\right|_{\mathsf{w}} - \left|U_v \cap \overline{U}_{\overline{x}}\right|_{\mathsf{w}} = \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. ■

The proof of the following claim is similar to the proof of Lemma 4.5.20, and is thus omitted.

Claim 4.5.21 *Let $v \in F_L$ and $\overline{x} = \pi(v)$. Under the assumptions of Section 4.5.2, if* $\deg(v) = \deg(\overline{x}) = 0$ *then* $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$.

Lemma 4.5.22 *Under the assumptions of Section 4.5.2, there does not exist a multiset $G \subseteq \mathsf{P}^{\mathsf{w}}$ of size Δ, such that $G \cap U = \emptyset$ and for all $p \in G$, it holds $\mathsf{d}(p, F) \leq 5\varrho$.*

Proof: Assume for the sake of contradiction that G exists. Then, we have $|U \cup G|_{\mathsf{w}} = |U|_{\mathsf{w}} + \Delta = n - m$, and moreover,

$$\begin{aligned} \nu(F, U \cup G) &\leq \nu(F,U) + \nu(F,G) \leq \nu(F,U) + 5\varrho\,|G|_{\mathsf{w}} = \nu(F,U) + 5\varrho\Delta \\ &\leq \frac{5}{3}(\nu(F,U) + 3\varrho\Delta) = \frac{5}{3}\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \leq 15\mathsf{opt}^{\mathsf{w}}, \end{aligned}$$

since $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \leq 9\mathsf{opt}^{\mathsf{w}}$, by Lemma 4.5.9. Namely, F is an acceptable solution, which contradicts Lemma 4.5.18. ■

Lemma 4.5.23 *Let $v \in F$ and $\overline{x} \in \overline{F}$. If $v\overline{x} \in \mathcal{E}$, then $\mathsf{d}(v, \overline{x}) \leq 4\varrho$, and furthermore, for all $p \in \overline{U}_{\overline{x}}$, it holds $\mathsf{d}(p, v) \leq 5\varrho$.*

Proof: Since $v\overline{x} \in \mathcal{E}$, there is a point q that is in both U_v and $\overline{U}_{\overline{x}}$. Therefore, we have $\mathsf{d}(q, v) \leq 3\varrho$ and $\mathsf{d}(q, \overline{x}) \leq \varrho$. By the triangle inequality, it holds $\mathsf{d}(v, \overline{x}) \leq \mathsf{d}(v, q) + \mathsf{d}(q, \overline{x}) \leq 4\varrho$. For an arbitrary point $p \in \overline{U}_{\overline{x}}$, we have that $\mathsf{d}(p, \overline{x}) \leq \varrho$, and as such, again by the triangle inequality, $\mathsf{d}(p, v) \leq \mathsf{d}(p, \overline{x}) + \mathsf{d}(\overline{x}, v) \leq \varrho + 4\varrho = 5\varrho$. ■

Lemma 4.5.24 *Under the assumptions of Section 4.5.2, there exists a heavy point $h \in \mathsf{P}^{\mathsf{w}}$ such that $h \notin U$ and furthermore, for all $v \in F$, it holds $\Delta \leq \mathsf{w}(h) \leq |U_v|_{\mathsf{w}}$.*

Proof: Consider an arbitrary heavy point h'. Since $|U|_{\mathsf{w}} < n - m$, it follows by the definition of U that h' appears either $\mathsf{w}(h')$ times or not at all in U. Note that the total weight of all the heavy points is $n - m$, namely $\mathsf{w}(\mathsf{C}_+) = n - m$. This implies (since $|U|_{\mathsf{w}} < n - m$) that there exists at least one heavy point that does not appear in U, and let h denote this point.

Assume, for the sake of contradiction, that $\mathsf{w}(h) \leq \Delta - 1$. Recall that $\mathsf{H} \subseteq \mathsf{C}_+$ is the set of k heaviest points and $|\mathsf{C}_+| = k + 1$. Thus, $n - m - |\mathsf{H}^{\mathsf{w}}|_{\mathsf{w}}$ is the weight of the heavy point with the least weight in C_+. As such, it holds that $n - m - |\mathsf{H}^{\mathsf{w}}|_{\mathsf{w}} \leq \mathsf{w}(h) \leq \Delta - 1$. By Observation 4.5.2 (i), we have

$$\begin{aligned} \mathcal{A}_m(\mathsf{H}, \mathsf{P}^{\mathsf{w}}, 3\varrho) &\leq \nu(\mathsf{H}, \mathsf{H}^{\mathsf{w}}) + 3\varrho(n - m - |\mathsf{H}^{\mathsf{w}}|_{\mathsf{w}}) \leq 0 + 3\varrho(\Delta - 1) \\ &\leq \nu(F, U) + 3\varrho\Delta - 3\varrho = \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - 3\varrho. \end{aligned} \tag{4.14}$$

On the other hand, since $F = \textsc{LocalSearch}(B, \mathsf{P}^{\mathsf{w}}, 3\varrho)$, where B is an arbitrary set of k facilities, it holds that $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - \varrho \leq \mathcal{A}_m(\mathsf{H}, \mathsf{P}^{\mathsf{w}}, 3\varrho)$, see Figure 4.2 (note that H is one of the candidate solutions considered by $\textsc{LocalSearch}$). Combining this inequality with Eq. (4.14), we obtain

$$\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - \varrho \leq \mathcal{A}_m(\mathsf{H}, \mathsf{P}^{\mathsf{w}}, 3\varrho) \leq \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - 3\varrho,$$

which is a contradiction.

Next, we prove the other inequality $\mathsf{w}(h) \leq |U_v|_{\mathsf{w}}$, for every $v \in F$. Assume for the sake of contradiction that $\mathsf{w}(h) > |U_v|_{\mathsf{w}}$. Let $M = (U \setminus U_v) \cup h^{\mathsf{w}}$. We have

$$|M|_{\mathsf{w}} = |U \setminus U_v|_{\mathsf{w}} + \mathsf{w}(h) = |U|_{\mathsf{w}} - |U_v|_{\mathsf{w}} + \mathsf{w}(h) \geq |U|_{\mathsf{w}} + 1,$$

since $h \notin U$ and $\mathsf{w}(h) > |U_v|_{\mathsf{w}}$. Now, note that

$$\nu(F - v + h, M) \leq \nu(F - v, U \setminus U_v) + \nu(\{h\}, h^{\mathsf{w}}) \leq \nu(F, U) + 0 = \nu(F, U).$$

If $|M|_{\mathsf{w}} \leq n - m$ then by Observation 4.5.2 (i), it holds that

$$\begin{aligned} \mathcal{A}_m(F - v + h, \mathsf{P}^{\mathsf{w}}, 3\varrho) &\leq \nu(F - v + h, M) + 3\varrho(n - m - |M|_{\mathsf{w}}) \\ &\leq \nu(F, U) + 3\varrho(n - m - |U|_{\mathsf{w}} - 1) = \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - 3\varrho, \end{aligned}$$

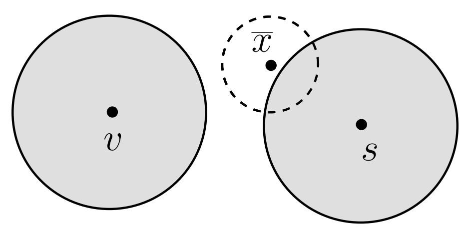
since $|M|_{\mathsf{w}} \geq |U|_{\mathsf{w}} + 1$. Now, arguing as above, this contradicts the local optimality of F, as $F - v + h$ is one of the possible solutions considered by LocalSearch, see Figure 4.2.

Otherwise, we have $|M|_{\mathsf{w}} > n - m$, and by Observation 4.5.2 (ii), it holds that

$$\begin{aligned} \mathcal{A}_m(F - v + h, \mathsf{P}^{\mathsf{w}}, 3\varrho) &\leq \nu(F - v + h, M) \leq \nu(F, U) \\ &\leq \nu(F, U) + 3\varrho(n - m - |U|_{\mathsf{w}} - 1) = \mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) - 3\varrho, \end{aligned}$$

since $n - m - |U|_{\mathsf{w}} - 1 \geq 0$ (implied by $|U|_{\mathsf{w}} < n - m$). Again, this is a contradiction to the local optimality of F. ■

Claim 4.5.25 *Let $v \in F_L$ and $\overline{x} = \pi(v)$. Under the assumptions of Section 4.5.2, if $\deg(v) = 0$, and there is a facility $s \in F$ such that $s \neq v$ and $s\overline{x} \in \mathcal{E}$, then $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$.*



Proof: Assume for the sake of contradiction that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. Since the degree of v is zero, we have $|U_v|_{\mathsf{w}} = \left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. By Lemma 4.5.24, we have $\Delta \leq |U_v|_{\mathsf{w}}$. It thus follows $\Delta < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$, and as such, there exists a subset $G \subseteq \overline{U}_{\overline{x}} \setminus U$ such that $|G|_{\mathsf{w}} = \Delta$. Furthermore, by Lemma 4.5.23, each point of $\overline{U}_{\overline{x}} \setminus U$ is within distance 5ϱ to s. Since $G \subseteq \overline{U}_{\overline{x}} \setminus U$, this implies that $\mathsf{d}(p, F) \leq \mathsf{d}(p, s) \leq 5\varrho$, for each $p \in G$. However, this contradicts Lemma 4.5.22. ■

The following claim will be useful in proving Claim 4.5.27 below.

Claim 4.5.26 *For $v \in F_L$ and $s \in F - v$ such that $\mathsf{d}(v, s) \leq 8\varrho$, it holds $\nu\left(F - v, U_v \cap \overline{U}\right) \leq 11\nu(v, U_v) + 2\nu\left(\overline{F}, \overline{U}\right)$.*

Proof: Consider $\overline{y} \in \overline{F}$ such that $U_v \cap \overline{U}_{\overline{y}}$ is not empty. For an arbitrary point $p \in U_v \cap \overline{U}_{\overline{y}}$, it holds $\mathsf{d}(v, p) \leq 3\varrho$ and $\mathsf{d}(\overline{y}, p) \leq \varrho$.

If $\mathsf{d}(\overline{y}, v) \geq 2\varrho$, then by the triangle inequality, we have that $\mathsf{d}(p,s) \leq \mathsf{d}(p,v) + \mathsf{d}(v,s) \leq 3\varrho + 8\varrho = 11\varrho$ and $\mathsf{d}(p,v) \geq \mathsf{d}(\overline{y},v) - \mathsf{d}(\overline{y},p) \geq 2\varrho - \varrho = \varrho$. In particular, for $p \in U_v \cap \overline{U}_{\overline{y}}$, we have $\nu(v,p) \geq \varrho$, and as such $\nu\big(v, U_v \cap \overline{U}_{\overline{y}}\big) \geq \varrho \left|U_v \cap \overline{U}_{\overline{y}}\right|_{\mathsf{w}}$. Since $s \in F - v$, we have

$$\nu\big(F - v, U_v \cap \overline{U}_{\overline{y}}\big) \leq \nu\big(s, U_v \cap \overline{U}_{\overline{y}}\big) \leq 11\varrho \left|U_v \cap \overline{U}_{\overline{y}}\right|_{\mathsf{w}} \leq 11\,\nu\big(v, U_v \cap \overline{U}_{\overline{y}}\big) = 11 \sum_{p \in U_v \cap \overline{U}_{\overline{y}}} \mathsf{d}(F,p).$$

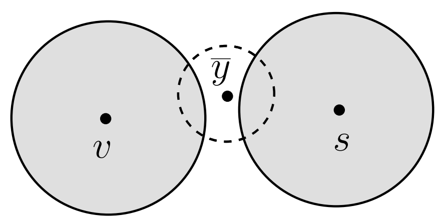
If $\mathsf{d}(\overline{y}, v) < 2\varrho$, then the distance between $\overline{y}$ and its nearest neighbor in F is less than 2ϱ, and as such, $\overline{y}$ is a prisoner of $\pi^{-1}(\overline{y})$. Note that $\pi^{-1}(\overline{y}) \neq v$, since otherwise, v captures $\overline{y}$, contradicting that $v \in F_L$. Claim 4.5.14 (ii) implies that

$$\nu\big(F - v, U_v \cap \overline{U}_{\overline{y}}\big) \leq \nu\big(\pi^{-1}(\overline{y}), U_v \cap \overline{U}_{\overline{y}}\big) \leq \sum_{p \in U_v \cap \overline{U}_{\overline{y}}} \big(\mathsf{d}(F,p) + 2\mathsf{d}(\overline{F},p)\big).$$

Combining these two cases, we obtain $\nu\big(F - v, U_v \cap \overline{U}_{\overline{y}}\big) \leq \sum_{p \in U_v \cap \overline{U}_{\overline{y}}} \big(11\mathsf{d}(F,p) + 2\mathsf{d}(\overline{F},p)\big)$. Summing the inequality over all facilities $\overline{y} \in \overline{F}$, we have that

$$\nu\big(F - v, U_v \cap \overline{U}\big) \leq \sum_{p \in U_v \cap \overline{U}} \big(11\mathsf{d}(F,p) + 2\mathsf{d}(\overline{F},p)\big) \leq 11\nu(v, U_v) + 2\nu\big(\overline{F}, \overline{U}\big)\,. \qquad \blacksquare$$

Claim 4.5.27 *Let $v \in F_L$ and $\overline{x} = \pi(v)$. Under the assumptions of Section 4.5.2, if there exists $\overline{y} \in \overline{F}$ and $s \in F$ such that $v\overline{y}, s\overline{y} \in \mathcal{E}$, then $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. Note that $s \neq v$, but it is possible that $\overline{y} = \overline{x}$.*



Proof: Assume for the sake of contradiction that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. By the triangle inequality and Lemma 4.5.23, it follows that $\mathsf{d}(v,s) \leq \mathsf{d}(v,\overline{y}) + \mathsf{d}(\overline{y},s) \leq 4\varrho + 4\varrho = 8\varrho$. And for any $q \in U_v$, we have $\mathsf{d}(v,q) \leq \mathsf{d}(q,v) + \mathsf{d}(v,s) \leq 3\varrho + 8\varrho = 11\varrho$.

(i) If $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \Delta$, then there exists a multiset $G \subseteq U_v \setminus \overline{U}$ such that $|G|_{\mathsf{w}} = \Delta$. Let $M' = (U \setminus U_v) \cup (U_v \cap \overline{U}) \cup G \cup (\overline{U}_{\overline{x}} \setminus U)$. Observe that M' is the union of the four disjoint sets. Indeed, $\overline{U}_{\overline{x}} \setminus U$ is disjoint from U, which contains the other three sets. It is easy to verify the other pairs of sets are also disjoint, using similar arguments. Therefore,

$$\begin{aligned}
|M'|_{\mathsf{w}} &= |U \setminus U_v|_{\mathsf{w}} + \left|U_v \cap \overline{U}\right|_{\mathsf{w}} + |G|_{\mathsf{w}} + \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}} \\
&= \big(|U|_{\mathsf{w}} - |U_v|_{\mathsf{w}}\big) + \big(|U_v|_{\mathsf{w}} - \left|U_v \setminus \overline{U}\right|_{\mathsf{w}}\big) + \Delta + \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}} \\
&= |U|_{\mathsf{w}} + \Delta - \left|U_v \setminus \overline{U}\right|_{\mathsf{w}} + \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}} > n - m,
\end{aligned}$$

since $|U|_{\mathsf{w}} + \Delta = n - m$ and $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$ (by assumption). Furthermore, for all $q \in G \subseteq U_v$, it holds that $\mathsf{d}(q,s) \leq 11\varrho$ and as such, $\nu(F - v, G) \leq \nu(s, G) \leq 11\varrho\,|G|_{\mathsf{w}} = 11\varrho\Delta$. Let $X = (U \setminus U_v) \cup$

$(U_v \cap \overline{U}) \cup G$, and by Claim 4.5.26, we have

$$\begin{aligned}
\Gamma &= \nu(F-v, X) = \nu(F-v, U \setminus U_v) + \nu\big(F-v, U_v \cap \overline{U}\big) + \nu(F-v, G) \\
&\le \nu(F-v, U \setminus U_v) + \big(11\nu(v, U_v) + 2\nu\big(\overline{F}, \overline{U}\big)\big) + 11\varrho\Delta \\
&\le 11\nu(F-v, U \setminus U_v) + 11\nu(v, U_v) + 11\varrho\Delta + 2\nu\big(\overline{F}, \overline{U}\big) \\
&= 11\nu(F, U) + 11\varrho\Delta + 2\nu\big(\overline{F}, \overline{U}\big) \le 11\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) + 2\mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) \\
&\le 11 \cdot 9\mathsf{opt}^{\mathsf{w}} + 2\mathsf{opt}^{\mathsf{w}} = 101\mathsf{opt}^{\mathsf{w}},
\end{aligned}$$

since $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \le 9\mathsf{opt}^{\mathsf{w}}$ and $\mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) \le \mathsf{opt}^{\mathsf{w}}$ by Lemma 4.5.9. Let $F' = F - v + \overline{x}$, we have

$$\begin{aligned}
\nu(F', M') &= \nu\big(F - v + \overline{x}, X \cup \big(\overline{U}_{\overline{x}} \setminus U\big)\big) \le \Gamma + \nu\big(\overline{x}, \overline{U}_{\overline{x}} \setminus U\big) \le 101\mathsf{opt}^{\mathsf{w}} + \nu\big(\overline{F}, \overline{U}\big) \\
&\le 101\mathsf{opt}^{\mathsf{w}} + \mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho) \le 102\mathsf{opt}^{\mathsf{w}}.
\end{aligned}$$

Namely, F' is an acceptable solution, which contradicts Lemma 4.5.18.

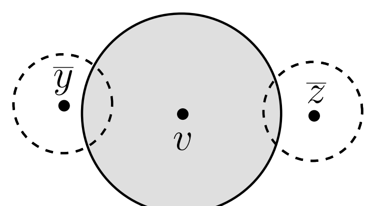
(ii) Consider the case that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \Delta$. By Lemma 4.5.24, there exists a heavy point h such that $h \notin U$ and $\mathsf{w}(h) \ge \Delta$. Let $M'' = U \cup h^{\mathsf{w}}$, and let $F'' = F - v + h$. It holds that $|M''|_{\mathsf{w}} = |U|_{\mathsf{w}} + \mathsf{w}(h) \ge |U|_{\mathsf{w}} + \Delta = n - m$. Set $G = U_v \setminus \overline{U}$, arguing as above, we have

$$\Gamma = \nu(F-v, U) = \nu\big(F - v, (U \setminus U_v) \cup (U_v \cap \overline{U}) \cup G\big) \le 101\mathsf{opt}^{\mathsf{w}},$$

and as such, $\nu(F'', M'') \le \Gamma + \nu(h, h^{\mathsf{w}}) \le 101\mathsf{opt}^{\mathsf{w}}$. Again, this implies that F'' is an acceptable solution, and this contradicts Lemma 4.5.18. ■

Claim 4.5.28 *Let $v \in F_L$ and $\overline{x} = \pi(v)$. Under the assumptions of Section 4.5.2, if there exists $\overline{y}, \overline{z} \in \overline{F}$ such that $v\overline{y}, v\overline{z} \in \mathcal{E}$ and $\deg(\overline{y}) = \deg(\overline{z}) = 1$ (namely, both $\overline{y}$ and $\overline{z}$ overlap only v), then $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \ge \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. Note that $\overline{y} \ne \overline{z}$, but it is possible that $\overline{y} = \overline{x}$ or $\overline{z} = \overline{x}$.*

Proof: Assume for the sake of contradiction that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} < \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$. Since both $\overline{y}$ and $\overline{z}$ overlap with only v, we have $\big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U = \big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U_v$. Also note that, by Lemma 4.5.23, every point in $\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}$ is within distance 5ϱ to v.



If $\left|\big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U\right|_{\mathsf{w}} \ge \Delta$ then there exists a subset $G \subseteq \big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U$ such that $|G|_{\mathsf{w}} = \Delta$. Since each point in G is within distance 5ϱ to v, this contradicts Lemma 4.5.22.

If $\left|\big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U\right|_{\mathsf{w}} < \Delta$ then we have $\nu\big(v, \big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U\big) \le 5\varrho \left|(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}) \setminus U\right|_{\mathsf{w}} < 5\varrho\Delta$, since every point in $\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}$ is within distance 5ϱ to v. Now, by Lemma 4.5.9, we have that $\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \le 9\mathsf{opt}^{\mathsf{w}}$, and as such,

$$\begin{aligned}
\nu\big(v, \overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) &= \nu\big(v, \big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \cap U_v\big) + \nu\big(v, \big(\overline{U}_{\overline{y}} \cup \overline{U}_{\overline{z}}\big) \setminus U_v\big) < \nu(v, U_v) + 5\varrho\Delta \\
&\le \nu(F, U) + 5\varrho\Delta \le \frac{5}{3}\mathcal{A}_m(F, \mathsf{P}^{\mathsf{w}}, 3\varrho) \le 15\mathsf{opt}^{\mathsf{w}}.
\end{aligned}$$

However, this contradicts Lemma 4.5.10. ■

Lemma 4.5.29 *Under the assumptions of Section 4.5.2, if $v \in F_L$ and $\overline{x} = \pi(v)$ then $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\overline{x}} \setminus U\right|_{\mathsf{w}}$.*

Proof: Consider the degrees of v and $\overline{x}$. There are six cases.

(i) If $\deg(v) = \deg(\overline{x}) = 0$, then the lemma holds by Claim 4.5.21.

(ii) If $\deg(v) = 0$ and $\deg(\overline{x}) \geq 1$, then the lemma holds by Claim 4.5.25.

(iii) If $\deg(v) = 1$, $\exists v\overline{y} \in \mathcal{E}$, and $\deg(\overline{y}) = 1$, then by definition, they match, which contradicts $v \in F_L$.

(iv) If $\deg(v) = 1$, $\exists v\overline{y} \in \mathcal{E}$, and $\deg(\overline{y}) > 1$, then the lemma holds by Claim 4.5.27.

(v) If $\deg(v) \geq 2$, $\exists v\overline{y}, v\overline{z} \in \mathcal{E}$, and $\deg(\overline{y}) = \deg(\overline{z}) = 1$, then the lemma holds by Claim 4.5.28.

(vi) If $\deg(v) \geq 2$, $\exists v\overline{y} \in \mathcal{E}$, and $\deg(\overline{y}) > 1$, then the lemma holds by Claim 4.5.27. ■

Lemma 4.5.17, Lemma 4.5.20, and Lemma 4.5.29 imply that $\left|U_v \setminus \overline{U}\right|_{\mathsf{w}} \geq \left|\overline{U}_{\pi(v)} \setminus U\right|_{\mathsf{w}}$ holds for every facility $v \in F$. As discussed, in Section 4.5.2, this implies Lemma 4.5.5.

Proof of Lemma 4.5.10

The proofs in this section depends only on the claims and lemmas preceding Lemma 4.5.10.

Lemma 4.5.30 *Under the assumptions of Section 4.5.2, there does not exist two heavy points h and h', a multiset $G \subseteq \mathsf{P}^{\mathsf{w}}$, and a facility $q \in \mathsf{P}^{\mathsf{w}}$, such that (i) $|G|_{\mathsf{w}} \geq \mathsf{w}(h) + \mathsf{w}(h')$, (ii) the multiset G excludes every heavy point in $\mathsf{C}_+ - h - h'$, and (iii) $\nu(q, G) \leq 15\mathsf{opt}^{\mathsf{w}}$.*

Proof: Assume for the sake of contradiction that they do exist. Let $B = \mathsf{C}_+ - h - h'$. Since $|\mathsf{C}_+| = k_+ = k+1$, we have $|B| = k-1$. It holds that $|B^{\mathsf{w}} \cup G|_{\mathsf{w}} = \mathsf{w}(B) + |G|_{\mathsf{w}} \geq \mathsf{w}(B) + \mathsf{w}(h) + \mathsf{w}(h') = \mathsf{w}(\mathsf{C}_+) = n - m$. Furthermore,

$$\nu(B + q, B^{\mathsf{w}} \cup G) \leq \nu(B, B^{\mathsf{w}}) + \nu(q, G) \leq 0 + 15\mathsf{opt}^{\mathsf{w}}.$$

Since $B + q \in \mathcal{H}$ and it is an acceptable solution, this contradicts Lemma 4.5.18. ■

Claim 4.5.31 *Under the assumptions of Section 4.5.2, the following holds:*

(i) There is at most one facility $\overline{x}$ in $\overline{F}$ such that $\overline{U}_{\overline{x}}$ partly-includes a heavy point.

(ii) There is no facility $\overline{x}$ in $\overline{F}$ such that $\overline{U}_{\overline{x}}$ includes two or more heavy points. (However, $\overline{U}_{\overline{x}}$ may include one heavy point and partly-include another heavy point.)

Proof: (i) Since $\overline{U} \subseteq \mathsf{P}^{\mathsf{w}}$ is the set of $n - m - \overline{\Delta}$ closest points to $\overline{F}$, and the inter-point distances of P are distinct, it follows that at most one heavy point can be "shattered" by $\overline{F}$.

(ii) Assume for the sake of contradiction that $\overline{U}_{\overline{x}}$ includes two heavy points h and h'. Let $G = \{h, h'\}^{\mathsf{w}}$. Since $\overline{U}_{\overline{x}}$ includes h and h', we have $G \subseteq \overline{U}_{\overline{x}}$, and as such $\nu(\overline{x}, G) \leq \nu(\overline{x}, \overline{U}_{\overline{x}}) \leq \nu(F, \overline{U}) \leq \mathsf{opt}^{\mathsf{w}}$, by Lemma 4.5.9. But this contradicts Lemma 4.5.30. ■

Lemma 4.5.32 *Let $\overline{x}, \overline{y} \in \overline{F}$ be two facilities. And let $\overline{U}_{\overline{x},\overline{y}} = \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}$ and $\overline{U}_{-\overline{x}-\overline{y}} = \overline{U} \setminus \overline{U}_{\overline{x},\overline{y}}$. Under the assumptions of Section 4.5.2, the following holds:*

(i) $\overline{U}_{-\overline{x}-\overline{y}}$ excludes at least two heavy points.

(ii) If h and h' are two heavy points excluded by $\overline{U}_{-\overline{x}-\overline{y}}$, then $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} \geq \mathsf{w}(h) + \mathsf{w}(h')$.

Proof: (i) By Claim 4.5.31, $\overline{U}_{-\overline{x}-\overline{y}}$ can only include at most $k-2$ heavy points, and may partly-include another heavy point. Since there are $k+1$ heavy points in total, there must be at least $(k+1)-(k-2)-1 = 2$ heavy points excluded by $\overline{U}_{-\overline{x}-\overline{y}}$.

(ii) Assume, for the sake of contradiction, that $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} < \mathsf{w}(h) + \mathsf{w}(h')$. Let $M = \overline{U}_{-\overline{x}-\overline{y}} \cup \{h, h'\}^{\mathsf{w}}$, and $\overline{F}' = \overline{F} - \overline{x} - \overline{y} + h + h'$. We have

$$\nu\left(\overline{F}', M\right) \leq \nu\left(\overline{F} - \overline{x} - \overline{y}, \overline{U}_{-\overline{x}-\overline{y}}\right) + \nu\left(\{h, h'\}, h^{\mathsf{w}} \cup h'^{\mathsf{w}}\right) \leq \nu\left(\overline{F}, \overline{U}\right) + 0 = \nu\left(\overline{F}, \overline{U}\right). \tag{4.15}$$

Furthermore, since $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} < \mathsf{w}(h) + \mathsf{w}(h')$, we have

$$|M|_{\mathsf{w}} = \left|\overline{U}_{-\overline{x}-\overline{y}}\right|_{\mathsf{w}} + \mathsf{w}(h) + \mathsf{w}(h') = \left|\overline{U}\right|_{\mathsf{w}} - \left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} + \mathsf{w}(h) + \mathsf{w}(h') > \left|\overline{U}\right|_{\mathsf{w}}.$$

If $|M|_{\mathsf{w}} \leq n - m$ then by Observation 4.5.2 (i) and Eq. (4.15), we have

$$\begin{aligned} \mathcal{A}_m(\overline{F}', \mathsf{P}^{\mathsf{w}}, \varrho) &\leq \nu\left(\overline{F}', M\right) + (n - m - |M|_{\mathsf{w}})\varrho \\ &< \nu\left(\overline{F}, \overline{U}\right) + (n - m - \left|\overline{U}\right|_{\mathsf{w}})\varrho = \mathcal{A}_m(\overline{F}, \mathsf{P}^{\mathsf{w}}, \varrho), \end{aligned}$$

since $|M|_{\mathsf{w}} > \left|\overline{U}\right|_{\mathsf{w}}$. This contradicts the optimality of $\overline{F}$.

If $|M|_{\mathsf{w}} > n - m$ then let $M' = \mathbf{N}_{n-m}(\overline{F}', M)$. Now, apply the above argument to $\overline{F}'$ and M', we similarly get a contradiction. ■

Lemma 4.5.10 (*restatement*) Under the assumptions of Section 4.5.2, for any $\overline{x}, \overline{y} \in \overline{F}$ and $q \in \mathsf{P}$, we have $\nu\left(q, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) \geq 15\mathsf{opt}^{\mathsf{w}}$.

Proof: Assume, for the sake of contradiction, that $\nu\left(q, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) < 15\mathsf{opt}^{\mathsf{w}}$. Let $\overline{U}_{\overline{x},\overline{y}} = \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}$ and $\overline{U}_{-\overline{x}-\overline{y}} = \overline{U} \setminus \overline{U}_{\overline{x},\overline{y}}$. There are several possibilities.

(i) $\overline{U}_{\overline{x},\overline{y}}$ includes two heavy points, h and h'. Let $G = h^{\mathsf{w}} \cup h'^{\mathsf{w}}$. Since $G \subseteq \overline{U}_{\overline{x},\overline{y}}$, we have $\nu(q, G) \leq \nu\left(q, \overline{U}_{\overline{x}} \cup \overline{U}_{\overline{y}}\right) \leq 15\mathsf{opt}^{\mathsf{w}}$ in this case. However, this is impossible, by Lemma 4.5.30.

(ii) $\overline{U}_{\overline{x},\overline{y}}$ includes one heavy point h, partly-includes another heavy point h', and excludes every other. In this case, h and h' are excluded by $\overline{U}_{-\overline{x}-\overline{y}}$, and as such, $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} \geq \mathsf{w}(h) + \mathsf{w}(h')$, by Lemma 4.5.32 (ii). Now, setting $G = \overline{U}_{\overline{x},\overline{y}}$, we have a contradiction, by Lemma 4.5.30.

(iii) $\overline{U}_{\overline{x},\overline{y}}$ excludes every heavy point except for h. In this case, h is excluded by $\overline{U}_{-\overline{x}-\overline{y}}$. In addition, By Lemma 4.5.32 (i), at least two heavy points are excluded by $\overline{U}_{-\overline{x}-\overline{y}}$, and as such, there must be another heavy point, say h', excluded by $\overline{U}_{-\overline{x}-\overline{y}}$. Now, by Lemma 4.5.32 (ii), we have $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} \geq \mathsf{w}(h) + \mathsf{w}(h')$. Now, setting $G = \overline{U}_{\overline{x},\overline{y}}$, we have a contradiction, by Lemma 4.5.30.

(iv) $\overline{U}_{\overline{x},\overline{y}}$ excludes every heavy point. In this case, by Lemma 4.5.32 (i), at least two heavy points, say h and h', are excluded by $\overline{U}_{-\overline{x}-\overline{y}}$, and as such, by Lemma 4.5.32 (ii), we have $\left|\overline{U}_{\overline{x},\overline{y}}\right|_{\mathsf{w}} \geq \mathsf{w}(h) + \mathsf{w}(h')$. Now, setting $G = \overline{U}_{\overline{x},\overline{y}}$, we have a contradiction, by Lemma 4.5.30.

■

4.6 Conclusions

In this chapter, we present the first efficient (i.e., polynomial time) constant-factor approximation algorithm for the k-median with outliers problem. A natural direction for future research is to extend the techniques used to other optimization problems with non-trivial global constraints, such as the capacitated k-median problem.

The new *successive local search* method, used in Section 4.2.2, is fairly general and should be applicable to other problems, since many combinatorial optimization problems can be reduced to their corresponding penalty versions. To use this method, however, it is crucial to bound the number of points that receive penalty. This is not easy and depends on the problem at hand.

Part II

Orienteering

Chapter 5

Introduction

Consider a traveling salesperson who has a fixed amount of gasoline (or time) and wants to maximize the number of customers visited under this constraint. This is an instance of the *orienteering problem* that requires us to design a network that visits a maximum number of points, subject to an upper bound on the total length of the network.

In this part of the thesis, we study the *rooted orienteering problem*: Given a set P of n points in the plane, a starting point $r \in P$, and a length constraint $\mathcal{B}$, one needs to find a path starting from r that visits as many points of P as possible and of length not exceeding $\mathcal{B}$. The effect of fixing the starting point is significant as far as approximation algorithms are concerned. Indeed, approximation algorithms for k-TSP extend easily to the *unrooted orienteering problem*, where there is no fixed starting point, while the approximation algorithm for the rooted orienteering problem is more challenging. The difficulty stems from the fact that an optimal path may visit a large number of points that lie in a small cluster at a distance nearly $\mathcal{B}$ from r, thus making it difficult to visit at least a large fraction of these points unless the path is very efficient [AMN98].

This problem is "dual" to the classical *k-TSP problem* [Aro98, Mit99, Gar05], which asks for a minimum length path visiting at least k points. Some other related problems include the *prize-collecting traveling salesman problem* and the *vehicle routing problem.* They arise from real world applications such as delivering goods to locations or assigning technicians to maintenance service jobs. A substantial amount of work on heuristics for these problems can be found in the operations research literature [TV02].

Arkin *et al.* [AMN98] were the first to design approximation algorithms for the rooted orienteering problem. They considered the rooted orienteering problem for points in the plane when the underlying network is a path, a cycle, or a tree. Their algorithms provide a 2-approximation for the rooted path orienteering problem, and a 2 (resp. 3) approximation when the networks considered are cycles (resp. trees). Blum *et al.* [BCK$^+$03] proposed the first constant-factor approximation algorithm for the rooted path orienteering problem when the points lie in a general metric space. Bansal *et al.* [BBCM04] improved the approximation factor to 3. Arkin *et al.* [AMN98] asked whether a better approximation is possible in Euclidean spaces.

One difficulty in assailing this problem is the relative lack of algorithmic tools to handle rigid budget constraints. Since the development of $(1 + \varepsilon)$-approximation algorithms for TSP [Aro98, Mit99], a large class of the problems that aim to minimize the tour length, subject to certain constraints on the points visited by the tour have been resolved. See the surveys by Mitchell [Mit00] and Arora [Aro03] for further information. In contrast, the behavior of the optimization problems that seek to maximize some function on the points visited, subject to constraints on the length of the path used or the timespans the points are being visited [BES05, BBCM04, CP05] are not as well understood.

The main idea in the previous approximation algorithms for the orienteering problem [AMN98, BCK+03, BBCM04] was to transform an approximation algorithm for the *rooted k-TSP problem* into an approximation algorithm for the orienteering problem. In particular, Blum *et al.* [BCK+03] formulated the notion of the *excess* for a path (which is defined to be the difference between the length of a path and the distance between the endpoints of the path), and then combined dynamic programming with the use of k-TSP to obtain an algorithm for orienteering in a metric space. These techniques were also implicitly used in the work of Arkin *et al.* [AMN98].

5.1 Our results

In Chapter 6, we extend the concept of the excess of a path into the $\mathfrak{u}$-*excess* of a path. It is (loosely) the difference in lengths between π and the best approximation to π by a polygonal line having $\mathfrak{u}$ vertices. (Therefore, the previous notion of excess is 2-excess in our notation.)

To obtain a PTAS for the orienteering problem, we revisit the rooted k-TSP problem in the plane, and show that Mitchell's algorithm [Mit99] computes an $(\varepsilon, \mathfrak{u})$-approximation for rooted k-TSP; that is, the algorithm outputs a rooted path of length $\leq \|\pi\| + \varepsilon \cdot \mathcal{E}_{\pi,\mathfrak{u}}$, where π is any path that starts from the root and visits k points, $\|\pi\|$ denotes the length of π, and $\mathcal{E}_{\pi,\mathfrak{u}}$ is the $\mathfrak{u}$-excess of π. Note that the quantity $\mathcal{E}_{\pi,\mathfrak{u}}$ might be smaller than $\|\pi\|$ by several orders of magnitude. Therefore, we show that Mitchell's algorithm provides a much tighter approximation for k-TSP than what was previously known. See Section 6.2.

In Section 6.3, we present an $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP in higher dimensions, by combining Mitchell's methods together with Arora's k-TSP algorithm [Aro98]. In particular, the algorithm top structure follows Mitchell's method. However, conceptually, whenever the algorithm "encounters" a dense window, it uses an easy extension of Arora's k-TSP algorithm.

Armed with the new approximation algorithm for k-TSP, it is now possible to reduce the orienteering problem to an instance of k-TSP. The PTAS for orienteering is presented in Section 6.4. In particular, in the plane, our algorithm computes, in $n^{O(1/\varepsilon)}$ time, a path that visits at least $(1-\varepsilon)k_{\text{opt}}$ points of P, where k_{opt} is the number of points visited by an optimal solution.

The results appeared in [CH06, CH07].

Chapter 6

$(1+\varepsilon)$ Approximation for Euclidean Orienteering

In this chapter, we present the first polynomial time $(1+\varepsilon)$ approximation algorithm for the rooted orienteering problem in Euclidean spaces. In Section 6.2, we revisit the rooted k-TSP problem in the plane, and show that Mitchell's algorithm [Mit99] computes an $(\varepsilon, \mathfrak{u})$-approximation for rooted k-TSP. We then extend the algorithm to higher dimensions in Section 6.3. We present the PTAS for the rooted orienteering problem in Section 6.4.

6.1 Definitions

Let $\pi = \langle p_1, p_2, \ldots, p_k \rangle$ be a path that visits k points of P, starting at p_1 and ending at p_k. The *length* of π is denoted by $\|\pi\| = \sum_{i=1}^{k-1} \|p_{i+1} - p_i\|$. More generally, for a collection S of segments, $\|S\|$ denotes the total length of segments in S. Let $1 = i_1 < \ldots < i_{\mathfrak{u}} = k$ be a sequence of $\mathfrak{u} \leq k$ integers. The path $\langle p_{i_1}, p_{i_2}, \ldots, p_{i_{\mathfrak{u}}} \rangle$ is a $\mathfrak{u}$-*skeleton* of π. The *optimal* $\mathfrak{u}$-*skeleton* of π is the $\mathfrak{u}$-skeleton of π with maximum total length, denoted by $\mathcal{S}^{\mathfrak{u}}_{\mathrm{opt}}(\pi)$. See Figure 6.1.

The $\mathfrak{u}$-*excess* of a path π is the difference between the length of π and its optimal $\mathfrak{u}$-skeleton, that is, $\mathcal{E}_{\pi,\mathfrak{u}} = \|\pi\| - \left\|\mathcal{S}^{\mathfrak{u}}_{\mathrm{opt}}(\pi)\right\|$. Note that the $\mathfrak{u}$-excess of π may be considerably smaller than the length of π.

Given a set P of n points and a starting point $r \in P$, the *rooted k-TSP problem* is to find a shortest path that visits k points of P starting at r. An $(\varepsilon, \mathfrak{u})$-*approximation* to the rooted k-TSP is a path ϕ that visit k points of P starting at r, such that the length of ϕ is no more than $\|\mathcal{T}\| + \varepsilon \cdot \mathcal{E}_{\mathcal{T},\mathfrak{u}}$, for any path $\mathcal{T}$ that visits k points of P starting at r.

Definition 6.1.1 (The rooted orienteering problem.) Given a set P of n points, a budget $\mathcal{B}$, and a starting point $r \in P$, the *rooted orienteering problem* is to find a path ω_{opt} that visits as many points of P as possible, under the constraint that the length of ω_{opt} is at most $\mathcal{B}$. Let k_{opt} denote the number of points visited by ω_{opt}. A $(1-\varepsilon)$-*approximation* to the rooted orienteering problem is a path ω (starting at r) that visits at least $(1-\varepsilon)k_{\mathrm{opt}}$ points of P, such that the length of ω is at most $\mathcal{B}$.

6.2 An $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP

In this section, we present an $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP. The algorithm is the k-TSP algorithm of Mitchell [Mit99], and our contribution is the new tighter analysis of its performance (see Section 6.2.3). In the following, we first review Mitchell's algorithm and then present our new improved analysis.

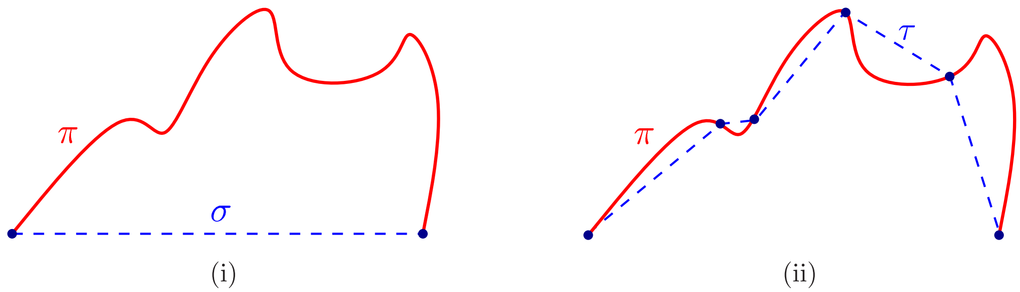


Figure 6.1: (i) The segment σ is the 2-skeleton of the path π. The 2-excess of π, namely $\mathcal{E}_{\pi,2}$, is the difference between the length of π and the length of σ. (ii) The polygonal line τ forms a 6-skeleton of π. The 6-excess of π, namely $\mathcal{E}_{\pi,6}$, is at most $\|\pi\| - \|\tau\|$.

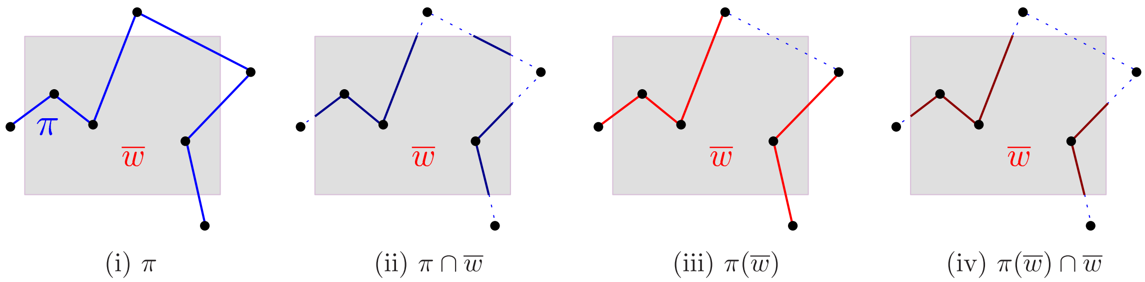


Figure 6.2: The different ways of clipping a path π to a window.

6.2.1 Preliminaries

In the following, m is a fixed constant. We assume, without loss of generality, that the points of P all have distinct x and y coordinates, and P is contained in an axis-parallel square $\mathcal{Q}$. Let π be a given path visiting k points of P.

Definition 6.2.1 A closed, axis-parallel rectangle $\overline{w}$ is a *window* if $\overline{w} \subseteq \mathcal{Q}$. The *extent* of a window $\overline{w}$ is the larger of the width and height of $\overline{w}$, and is denoted by $\Delta_{\overline{w}}$. Let $\pi(\overline{w})$ denote the subset of π consisting of the union of segments of π having at least one endpoint inside (or on the boundary of) $\overline{w}$. Given a collection S of segments, we slightly abuse notations and denote the set of segments of S clipped to $\overline{w}$ by $S \cap \overline{w}$. See Figure 6.2.

A line ℓ is a *cut* for π, with respect to $\overline{w}$, if ℓ is a horizontal or vertical line and ℓ intersects $\overline{w}$; ℓ is an *m-perfect cut* for π, with respect to $\overline{w}$, if ℓ intersects the segments of $\pi(\overline{w}) \cap \overline{w}$ at most m times.

Definition 6.2.2 The combinatorial type of a window $\overline{w}$ with respect to π is the subset of P inside (or on the boundary of) $\overline{w}$ and a listing, for each of the four sides of $\overline{w}$, of the identities of the line segments of $\pi(\overline{w})$ that intersect it. (In particular, if a segment of π intersects $\overline{w}$ but both its endpoints are outside $\overline{w}$, then the segment is not considered in the combinatorial type of $\overline{w}$.) We say that $\overline{w}$ is a *minimal window* if there is no window $\overline{w}'$ that is strictly contained in $\overline{w}$ with the same combinatorial type as $\overline{w}$.

For a minimal window $\overline{w}$, if there is no m-perfect cut for π, with respect to $\overline{w}$, then it is *m-dense*. Namely, any horizontal or vertical line that intersects $\overline{w}$ has more than m intersection points with $\pi(\overline{w}) \cap \overline{w}$.

Given a window $\overline{w}$, Mitchell [Mit99] described how to "shrink" $\overline{w}$ into a minimal window by "pinning" all four sides of $\overline{w}$. It is not hard to see that the number of all possible minimal windows is $O(n^4)$, since

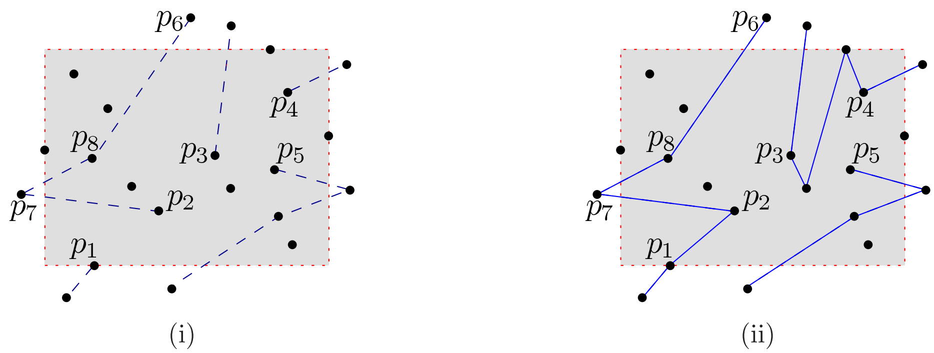


Figure 6.3: (i) An instance of the WINDOWTSP problem. The segments specify how the solution crosses the boundary of $\overline{w}$. The connectivity constraints are as follows: p_1 is required to connect to p_2 (possibly via other points within $\overline{w}$), p_3 is required to connect to p_4 (possibly via other points within $\overline{w}$), and p_5 is the starting point r of the path (namely, the degree of p_5 is 1). The multi-paths are required to visit 9 points in the window. (ii) A possible solution.

(intuitively) it has four degrees of freedom.

Claim 6.2.3 *If a window $\overline{w}$ is m-dense then* $\|\pi(\overline{w}) \cap \overline{w}\| \geq m \cdot \Delta_{\overline{w}}$.

Proof: Assume, without loss of the generality, that the width of $\overline{w}$ is greater than the height of $\overline{w}$, and the interval $[x_1, x_2]$ is the projection of $\overline{w}$ onto the x-axis. Let $f(\alpha)$ denote the number of segments of $\pi(\overline{w})$ within $\overline{w}$ that intersects the vertical line $x = \alpha$. By the density of $\overline{w}$, $f(x) > m$ for $x \in [x_1, x_2]$. The total length of the segments of $\pi(\overline{w}) \cap \overline{w}$ is lower bounded by the integral of $f(x)$ over $[x_1, x_2]$, which in turn is lower bounded by $m(x_2 - x_1) = m \cdot \Delta_{\overline{w}}$. ■

6.2.2 Review of the k-TSP algorithm

The "new" $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP is the algorithm of Mitchell [Mit99] for k-TSP, and we review it here only for the sake of completeness. We remind that the reader that m is a fixed (constant) number.

A problem instance of WINDOWTSP consists of: (i) a (minimal) window $\overline{w}$ that contains at least one point of P, with its boundaries determined by (up to) four points of P, (ii) an integer $h \geq 0$, indicating how many points interior to $\overline{w}$ should be visited, (iii) boundary information specifying at most m crossing segments (each determined by a pair of points of P, one interior to or on the boundary of $\overline{w}$, another outside $\overline{w}$) for each side of the boundary of $\overline{w}$, and (iv) connectivity constraints, indicating which pairs of crossing segments are required to be connected within $\overline{w}$. The solution to the WINDOWTSP problem is a set of (hopefully short) paths inside window $\overline{w}$ such that: (i) all of the boundary constraints are satisfied, (ii) all of the connectivity constraints within $\overline{w}$ are satisfied, and (iii) h points of P are visited by the paths inside $\overline{w}$. See Figure 6.3.

Clearly, the rooted k-TSP problem can be formulated as a WINDOWTSP instance consisting of a bounding box $\mathcal{Q}$ of P, a parameter k, empty boundary information, and connectivity constraints requiring that k points of P inside $\mathcal{Q}$ must be connected by a single path, with r as an endpoint of the path.

The recursive algorithm for WINDOWTSP works as follows. If the window $\overline{w}$ has at most m points of P in its interior, then the problem is solved by enumeration of all possible solutions. Otherwise, the

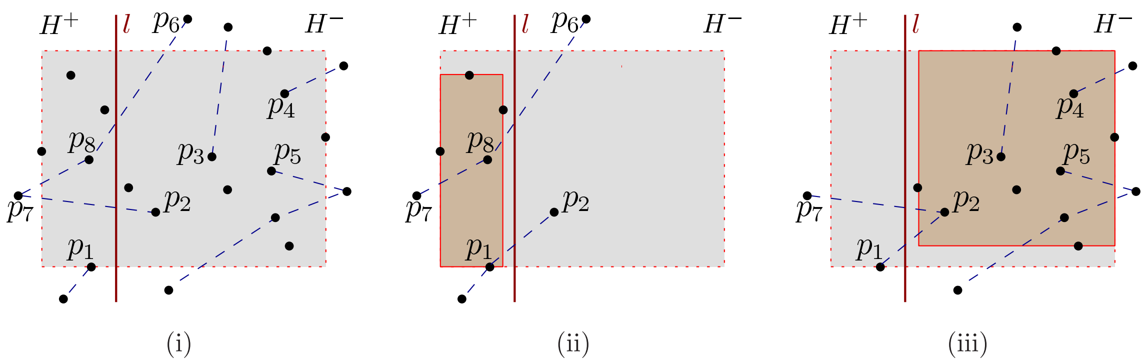


Figure 6.4: Performing a cut on the WINDOWTSP instance of Figure 6.3. (i) A cut l divides the window $\overline{w}$ into smaller windows $\overline{w} \cap H^+$ and $\overline{w} \cap H^-$. (ii) The minimal window for $\overline{w} \cap H^+$. The segments represent the crossing boundary information for the new window. In particular, the segment p_1p_2 is introduced when "guessing" the boundary information along the cut l before the recursive call. The other segments intersecting the boundary are inherited from the original instance. (iii) The minimal window for $\overline{w} \cap H^-$.

algorithm tries all possible cuts for the current window recursively, enumerating over all the possible choices of valid boundary information along this cut and computing the cheapest option. If there is no m-perfect cut then the algorithm performs a cut and reduces the intersection by introducing bridges; see Remark 6.2.4 below. For our analysis, we only care whether a cut used by the algorithm is an m-perfect cut, or is it a more complicated cut.

When a cut divides a window into two smaller windows, we need to "shrink" those two windows into minimal windows. In particular, segments that just pass through a window are ignored during the shrinking. This is a small but important technicality. See Figure 6.4.

Remark 6.2.4 The algorithm of Mitchell [Mit99] also introduces bridges (close to, or) on the boundary of the window $\overline{w}$, where a bridge is a vertical or horizontal segment. To simplify our exposition, we ignored those bridges in describing the algorithm. Of course, for a correct working implementation those bridges are necessary. See [Mit99] for full details. See also Remark 6.2.10 below.

6.2.3 Analysis of the algorithm

The key observation in our analysis is that the approximation algorithm does not introduce any error when a cut is m-perfect. Thus, the error is introduced only when the algorithm works inside an m-dense window, but such windows have high "excess".

Definition 6.2.5 For a set S of segments, let $I_x(S)$ denote the projection of S to x-axis; namely, $I_x(S)$ is the set of all points α (on the x-axis), such that the vertical line $x = \alpha$ intersects the segments of S. Let $\mathrm{len}_x(S)$ denote the total length of $I_x(S)$. Note that $I_x(S)$ is a set of (disjoint) intervals on the real line, and $\mathrm{len}_x(S)$ is the total length of these intervals. We define $I_y(S)$ and $\mathrm{len}_y(S)$ in a similar fashion.

Definition 6.2.6 For a window $\overline{w}$ and a path π, the *surplus* of π in $\overline{w}$ is

$$\rho(\overline{w}, \pi) = \|\pi \cap \overline{w}\| - \sqrt{(\mathrm{len}_x(\pi \cap \overline{w}))^2 + (\mathrm{len}_y(\pi \cap \overline{w}))^2}.$$

It is easy to verify that the surplus is always non-negative. See Figure 6.5.

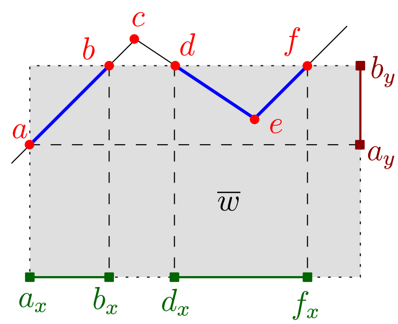


Figure 6.5: Illustrating the intersection of a polygonal line $\pi = \langle a, b, c, d, e, f \rangle$ with a window $\overline{w}$. The set $I_x(\pi \cap \overline{w})$ consists of the segments $a_x b_x$ and $d_x f_x$; the set $I_y(\pi \cap \overline{w})$ consists of segment $a_y b_y$. The surplus of π in $\overline{w}$ is $\|a-b\| + \|d-e\| + \|e-f\| - \sqrt{(\|a_x - b_x\| + \|d_x - f_x\|)^2 + \|a_y - b_y\|^2}$.

Lemma 6.2.7 *If $X, Y, X_1, \ldots, X_n, Y_1, \ldots, Y_n$ are non-negative real numbers such that*

$$\sum_{i=1}^{n} X_i \geq X \quad \text{and} \quad \sum_{i=1}^{n} Y_i \geq Y,$$

then $\sum_{i=1}^{n} \sqrt{X_i^2 + Y_i^2} \geq \sqrt{X^2 + Y^2}$.

Proof: Let q_i be the point $(\sum_{j=1}^{i} X_j, \sum_{j=1}^{i} Y_j)$ in the plane, for $1 \leq i \leq n$; and let $q_0 = (0, 0)$. Consider the path $\pi = \langle q_0, q_1, \ldots, q_n \rangle$. Clearly, we have that

$$\begin{aligned} \|\pi\| &= \sum_{i=1}^{n} \sqrt{X_i^2 + Y_i^2} = \sum_{i=1}^{n} \|q_i - q_{i-1}\| \geq \|q_n - q_0\| \\ &= \sqrt{\left(\sum_{i=1}^{n} X_i\right)^2 + \left(\sum_{i=1}^{n} Y_i\right)^2} \geq \sqrt{X^2 + Y^2}, \end{aligned}$$

as required. ■

Lemma 6.2.8 *Let $\overline{\mathsf{D}}$ be a set of interior disjoint windows (inside $\mathcal{Q}$), and let π be a polygonal path inside $\mathcal{Q}$. We have that $\mathcal{E}_{\pi,2} \geq \sum_{\overline{w} \in \overline{\mathsf{D}}} \left(\|\pi \cap \overline{w}\| - \sqrt{2} \Delta_{\overline{w}} \right)$.*

Proof: Let Ψ be a decomposition of $\mathcal{Q}$ into interior disjoint axis-parallel rectangles such that Ψ contains all of the rectangles of $\overline{\mathsf{D}}$. Let $X = \mathrm{len}_x(\pi \cap \mathcal{Q})$ and $Y = \mathrm{len}_y(\pi \cap \mathcal{Q})$. For a window $\overline{w}$, let $X_{\overline{w}} = \mathrm{len}_x(\pi \cap \overline{w})$ and $Y_{\overline{w}} = \mathrm{len}_y(\pi \cap \overline{w})$. Clearly, $X \leq \sum_{\overline{w} \in \Psi} X_{\overline{w}}$ and $Y \leq \sum_{\overline{w} \in \Psi} Y_{\overline{w}}$, since $I_x(\pi) = \bigcup_{\overline{w} \in \Psi} I_x(\pi \cap \overline{w})$ and $I_y(\pi) = \bigcup_{\overline{w} \in \Psi} I_y(\pi \cap \overline{w})$. Let s and t be the two endpoints of π. We have that

$$\mathcal{E}_{\pi,2} = \|\pi\| - \|s - t\| = \|\pi\| - \sqrt{\mathrm{len}_x(st)^2 + \mathrm{len}_y(st)^2} \geq \|\pi\| - \sqrt{X^2 + Y^2},$$

since $\mathrm{len}_x(st) \leq X$ and $\mathrm{len}_y(st) \leq Y$. On the other hand, by Lemma 6.2.7, we get $\sqrt{X^2+Y^2} \leq \sum_{\overline{w}\in\Psi} \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2}$, since $\sum_{\overline{w}\in\Psi} X_{\overline{w}} \geq X$ and $\sum_{\overline{w}\in\Psi} Y_{\overline{w}} \geq Y$. Therefore,

$$\begin{aligned}
\mathcal{E}_{\pi,2} &\geq \|\pi\| - \sqrt{X^2+Y^2} \geq \|\pi\| - \sum_{\overline{w}\in\Psi} \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2} = \sum_{\overline{w}\in\Psi} \left(\|\pi \cap \overline{w}\| - \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2} \right) \\
&= \sum_{\overline{w}\in\Psi} \rho(\overline{w},\pi) = \sum_{\overline{w}\in\overline{\mathsf{D}}} \rho(\overline{w},\pi) + \sum_{\overline{w}\in\Psi\setminus\overline{\mathsf{D}}} \rho(\overline{w},\pi) \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \rho(\overline{w},\pi),
\end{aligned}$$

because the surplus $\rho(\overline{w},\pi)$ is always non-negative. Now, since

$$\rho(\overline{w},\pi) = \|\pi \cap \overline{w}\| - \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2} \geq \|\pi \cap \overline{w}\| - \sqrt{\Delta_{\overline{w}}^2 + \Delta_{\overline{w}}^2} = \|\pi \cap \overline{w}\| - \sqrt{2}\Delta_{\overline{w}},$$

we obtain $\mathcal{E}_{\pi,2} \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \rho(\overline{w},\pi) \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \big(\|\pi \cap \overline{w}\| - \sqrt{2}\Delta_{\overline{w}}\big)$, as claimed. ■

Theorem 6.2.9 *Let $\pi = \langle p_1, p_2, \ldots, p_k\rangle$ be an arbitrary path that visits k points of P, and let $\mathfrak{u} \geq 2$ be an arbitrary fixed integer. One can compute, in $n^{O(\mathfrak{u})}$ time, a path that starts at p_1 and visits k points of P, and its length is at most $\|\pi\| + \mathcal{E}_{\pi,\mathfrak{u}}/\mathfrak{u}$.*

Proof: Set $m = \lceil 2\sqrt{2}\mathfrak{u} \rceil$, and use the algorithm presented above. The running time bound follows readily. Thus, we only need to argue that the path computed is indeed within the claimed bound on the length.

Thus, consider the (conceptual) execution of the recursive algorithm over the path π, performing the recursive calls according to π. Specifically, let $\overline{w}$ be a *minimal* window that is visited by the recursive algorithm when applied to π. If an m-perfect cut (for π) exists with respect to $\overline{w}$, then we use it to cut the window $\overline{w}$ into two parts and proceed recursively on each side of the cut. If such an m-perfect cut does not exist for $\overline{w}$ (that is, $\overline{w}$ is m-dense), then we (conceptually) stop, and use the results returned by the recursive call on this window. We claim that for these specific choices, the recursive algorithm computes a path σ, such that $\|\sigma\|$ is as required. Since the recursive algorithm returns a path no longer than σ, this would imply the theorem.

Let $\overline{\mathsf{D}}$ be the set of m-dense windows (which by the algorithm execution are minimal windows) visited by the algorithm when applied to π. Let $\mathcal{S} = \mathcal{S}^{\mathfrak{u}}_{\mathrm{opt}}(\pi)$ be an optimal $\mathfrak{u}$-skeleton for π, and let $\pi_1, \ldots, \pi_{\mathfrak{u}-1}$ be the breakup of π into subpaths by the vertices of $\mathcal{S}$. By Lemma 6.2.8, we have that

$$\begin{aligned}
\mathcal{E}_{\pi,\mathfrak{u}} &= \sum_{j=1}^{\mathfrak{u}-1} \mathcal{E}_{\pi_j,2} \geq \sum_{j=1}^{\mathfrak{u}-1} \sum_{\overline{w}\in\overline{\mathsf{D}}} \left(\|\pi_j \cap \overline{w}\| - \sqrt{2}\Delta_{\overline{w}} \right) = \sum_{\overline{w}\in\overline{\mathsf{D}}} \sum_{j=1}^{\mathfrak{u}-1} \left(\|\pi_j \cap \overline{w}\| - \sqrt{2}\Delta_{\overline{w}} \right) \\
&= \sum_{\overline{w}\in\overline{\mathsf{D}}} \left(\|\pi \cap \overline{w}\| - \sqrt{2}(\mathfrak{u}-1)\Delta_{\overline{w}} \right) \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \frac{\|\pi \cap \overline{w}\|}{2},
\end{aligned}$$

since $\|\pi \cap \overline{w}\| \geq m\Delta_{\overline{w}}$, for each $\overline{w} \in \overline{\mathsf{D}}$ (by Claim 6.2.3), and $m = \lceil 2\sqrt{2}\mathfrak{u} \rceil \geq 2\sqrt{2}\mathfrak{u}$. Now, note that $\pi(\overline{w}) \cap \overline{w}$ is a subset of $\pi \cap \overline{w}$, and henceforth it holds

$$\mathcal{E}_{\pi,\mathfrak{u}} \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \frac{\|\pi \cap \overline{w}\|}{2} \geq \sum_{\overline{w}\in\overline{\mathsf{D}}} \frac{\|\pi(\overline{w}) \cap \overline{w}\|}{2}. \tag{6.1}$$

For an m-dense window $\overline{w} \in \overline{\mathsf{D}}$, the path σ output by the algorithm (when applied to π) inside $\overline{w}$ is of length $\leq (1 + 1/m) \cdot \|\pi(\overline{w}) \cap \overline{w}\|$, as this is the performance guarantee provided by Mitchell's analysis [Mit99]. Namely, the error introduced by the approximation inside $\overline{w}$ is bounded by $\|\pi(\overline{w}) \cap \overline{w}\| / m$. For windows (visited by the algorithm when applied to π) that are not m-dense, the path σ within them is identical to the path π. Thus, for the path σ, it follows from Eq. (6.1) that

$$\|\sigma\| - \|\pi\| \leq \sum_{\overline{w} \in \overline{\mathsf{D}}} \frac{\|\pi(\overline{w}) \cap \overline{w}\|}{m} = \frac{2}{m} \sum_{\overline{w} \in \overline{\mathsf{D}}} \frac{\|\pi(\overline{w}) \cap \overline{w}\|}{2} \leq \frac{2\mathcal{E}_{\pi,\mathfrak{u}}}{m} < \frac{\mathcal{E}_{\pi,\mathfrak{u}}}{\mathfrak{u}},$$

since $m \geq 2\sqrt{2}\mathfrak{u}$. ■

It is possible to prove Lemma 6.2.8 and Theorem 6.2.9 directly, by arguing that the skeleton can be replaced by an alternative skeleton that is longer and is still shorter, by the excess in the dense windows, than an optimal path. (Since excess is a global property that is not directly defined for windows, the resulting argument in somewhat more complicated.) We provide the more technical proof above, since it brings to the forefront the notion of surplus. Note that the surplus is decomposition sensitive, as such, it might be much smaller than the excess. Therefore, the analysis of Theorem 6.2.9 is probably loose, as the bound on the error depends solely on the surplus in the dense windows.

Remark 6.2.10 As mentioned in Remark 6.2.4, we ignored the use of bridges in describing Mitchell's algorithm [Mit99]. Our analysis implies that the use of those bridges is restricted only to dense windows, where all we need is the performance guarantees already provided by Mitchell's analysis. In particular, for those dense windows, we can also use Arora's algorithm. This is the main insight we use in extending our algorithm to higher dimensions.

6.3 An $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP in $\mathbb{R}^d$

In this section, we present an $(\varepsilon, \mathfrak{u})$-approximation algorithm for k-TSP in higher dimensions, by combining Mitchell's methods together with Arora's k-TSP algorithm [Aro98]. Throughout the section, we are concerned with $\mathbb{R}^d$, where $d > 2$ is a fixed constant.

Let $\mathcal{Q}$ be an axis-parallel d-dimensional hypercube that contains the point set P. In the following, m is a fixed constant, π is a given path visiting k points of P. The following definitions are analogous to the ones in Section 6.2.

Definition 6.3.1 A closed, axis-parallel d-dimensional box $\overline{w}$ is a *window* if $\overline{w} \subseteq \mathcal{Q}$. The *extent* of a window $\overline{w}$ is the largest side length of $\overline{w}$, and is denoted by $\Delta_{\overline{w}}$.

A $(d-1)$-dimensional hyperplane ℓ is a *cut* for π, with respect to $\overline{w}$, if ℓ is axis-parallel and ℓ intersects $\overline{w}$; ℓ is an *m-perfect cut* for π, with respect to $\overline{w}$, if ℓ intersects the segments of $\pi(\overline{w}) \cap \overline{w}$ at most m times.

Definition 6.3.2 The combinatorial type of a window $\overline{w}$ with respect to π is the subset of P inside (or on the boundary of) $\overline{w}$ and a listing, for each facet of $\overline{w}$, of the identities of the line segments of $\pi(\overline{w})$ that intersect it. We say that $\overline{w}$ is a *minimal window* if there is no window $\overline{w}'$ that is strictly contained in $\overline{w}$ with the same combinatorial type as $\overline{w}$.

For a minimal window $\overline{w}$, if there is no m-perfect cut for π, with respect to $\overline{w}$, then it is *m-dense.* Namely, any axis-parallel $(d-1)$-dimensional hyperplane that intersects $\overline{w}$ has more than m intersection points with $\pi(\overline{w}) \cap \overline{w}$.

The following claim is an immediate extension of Claim 6.2.3.

Claim 6.3.3 *If a window $\overline{w}$ is m-dense then* $\|\overline{w} \cap \pi\| \geq m \cdot \Delta_{\overline{w}}$.

To bootstrap our algorithm, we need a $(1+\varepsilon)$-approximation algorithm for the WindowTSP problem in $\mathbb{R}^d$. To this end, note that the WindowTSP problem seeks a set of $O(md)$ paths (with prespecified endpoints) that collectively visits a prespecified number of points inside the given window; it is not hard to adapt Arora's technique to solve the WindowTSP problem in $n^{O(md)} \cdot O(m \log n)^{(m\sqrt{d})^{O(d)}}$ time, such that the solution has a total length $\leq (1 + 1/m)L(\overline{w})$, where $L(\overline{w})$ denotes the length of an optimal solution inside the specified window $\overline{w}$. (This problem can be solved even faster, but it has no impact on the overall performance of our algorithm.) The required adaption is straightforward and we omit the tedious but easy details; see [Aro98, AK03]. We denote this subroutine by kDenseAprxTSP.

For an instance of the WindowTSP problem, the algorithm works as follows. If $\overline{w}$ has at most m points of P in its interior, then the subproblem is solved by brute force. Otherwise, the algorithm chooses the smaller value returned by the following two options.

(a) Use kDenseAprxTSP to solve this problem, providing a solution with total length at most $(1 + 1/m)L(\overline{w})$, where $L(\overline{w})$ denotes the length of an optimal solution inside $\overline{w}$.

(b) Solve the problem recursively, optimizing over all choices associated with an m-perfect cut of window $\overline{w}$. (As in the $\mathbb{R}^2$ case, before performing the recursive calls, we need to shrink the windows formed by the cut into minimal windows.)

   (i) There are $O(d \cdot n^2)$ choices for a cut. More specifically, there are d choices of (axial) directions; and we can always let the cut pass through either a point of P or an intersection point between $\pi(\overline{w})$ and the boundary of $\overline{w}$. Since $\pi(\overline{w})$ is a subset of the set of $\binom{n}{2}$ possible segments (namely, all segments connecting a pair of points of P), it follows that there are $O(n^2)$ possible intersection points between $\pi(\overline{w})$ and the boundary of $\overline{w}$.

   (ii) There are $O(k)$ choices of the number of points visited in new subproblems, subject to the requirement that the total number of points visited within the two subproblems is equal to the number specified in the given instance.

   (iii) There are $O(n^{2m})$ choices of new boundary information on the cut. Specifically, we select $\leq m$ segments (each determined by a pair of points of P) that cross the cut. We require that the boundary information of the new subproblems be consistent with the boundary information of the given instance.

   (iv) There are a constant number of choices (since m and d are fixed constants) of connectivity constraints for the two new subproblems determined by the cut, subject to the requirement that these constraints be consistent with the constraints of the given instance.

Let kTSPAprxAlg denote this recursive algorithm. One can easily use memoization to turn it into an efficient dynamic programming algorithm. There are $O\big(k \cdot n^{2d} \cdot (n^{2m})^{2d}\big) = O\big(k\, n^{(4m+2)d}\big)$ possible subproblems, since there are $O(k)$ choices for the number of points that should be visited within $\overline{w}$, $O(n^{2d})$ choices of $\overline{w}$, and $O(n^{2m})$ choices of crossing segments on each of the $2d$ facets of $\overline{w}$. (The number of possible connectivity constraints is a constant since m and d are fixed constants.)

Remark 6.3.4 Before analyzing this algorithm, observe that it can be viewed as the combination of Mitchell's method [Mit99] with Arora's k-TSP algorithm [Aro98]. Specifically, the algorithm top structure follows Mitchell's method. However, conceptually, whenever the algorithm "encounters" a dense window, it uses kDenseAprxTSP (which is an easy extension of Arora's k-TSP algorithm).

To see why we had to modify Mitchell's algorithm, observe that the algorithm in Section 6.2 cannot be used in higher dimensions directly, because part of Mitchell's k-TSP algorithm relies on a crucial property of m-guillotine subdivisions in the plane. Namely, it introduces (and accounts for the additional length of) bridges on the path to decrease the interaction of the path with the outside world when considering dense windows. It is not known how to extend this directly to higher dimensions. However, the need for bridges arises only when a window is dense. In a dense window (in higher dimensions) we can circumvent this issue altogether by using Arora's algorithm (namely kDenseAprxTSP). Similarly, using Arora's algorithm on its own does not suffice here, since it introduces errors (by deflecting paths through "portals") even in windows which are not dense.

Analysis. To analyze the algorithm, we extend the definition of surplus (see Definition 6.2.6) to higher dimensions in a natural way. The following lemma is the analog of Lemma 6.2.8.

Lemma 6.3.5 *Let $\overline{\mathsf{D}}$ be a set of interior disjoint windows (inside $\mathcal{Q}$), and let π be a polygonal path inside $\mathcal{Q}$. We have that $\mathcal{E}_{\pi,2} \geq \sum_{\overline{w}\in\overline{\mathsf{D}}}\Big(\|\pi \cap \overline{w}\| - \sqrt{d}\Delta_{\overline{w}}\Big)$.*

The following theorem is similar to Theorem 6.2.9.

Theorem 6.3.6 *Let $\pi = \langle p_1, p_2, \ldots, p_k \rangle$ be an arbitrary path that visits k points of P, and let $\mathfrak{u} \geq 2$ be an arbitrary fixed integer. One can compute, in $n^{O(\mathfrak{u}d\sqrt{d})} \cdot (\mathfrak{u}\sqrt{d}\log n)^{(\mathfrak{u}d)^{O(d)}}$ time, a path that starts at p_1 and visits k points of P, and its length is at most $\|\pi\| + \mathcal{E}_{\pi,\mathfrak{u}}/\mathfrak{u}$.*

Proof: Set $m = \left\lceil 2\sqrt{d} \cdot \mathfrak{u} \right\rceil$. Observe that the algorithm kTSPAprxAlg uses the algorithm kDenseAprxTSP inside the dense windows, which provides the required approximation guarantee. The argument now follows the proof of Theorem 6.2.9 (almost) verbatim, and is thus omitted. ∎

Note, that the algorithm in this section also works for the planar case (namely $d = 2$).

6.4 A PTAS for orienteering

Next, we apply the algorithm of Theorem 6.3.6 to the rooted orienteering problem.

Lemma 6.4.1 *Given a set P of n points in $\mathbb{R}^d$, a budget $\mathcal{B}$, and a root $r = p_1$, let $\pi^*_{\mathrm{opt}} = \langle p_1, p_2, \ldots, p_k \rangle$ be an optimal rooted orienteering path starting at r with budget $\mathcal{B}$. One can compute, in $n^{O(d\sqrt{d}/\varepsilon)} \cdot (\sqrt{d}\log n/\varepsilon)^{(d/\varepsilon)^{O(d)}}$ time, a path such that it starts at r and visits at least $(1-\varepsilon)k$ points of P, and its length is at most $\mathcal{B}$.*

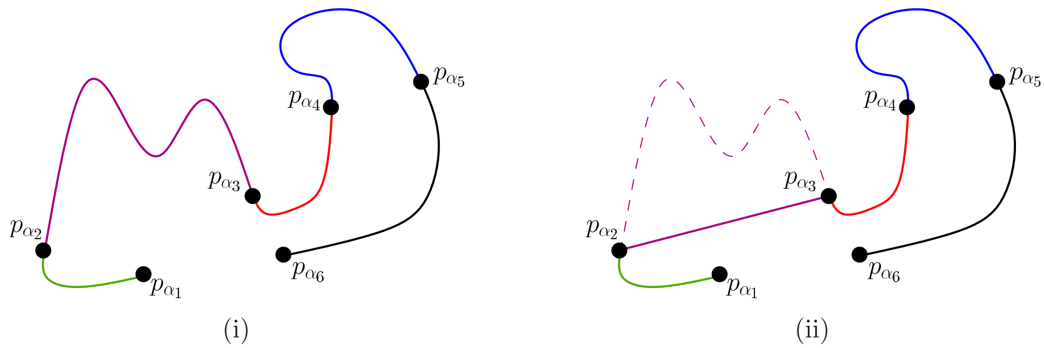


Figure 6.6: (i) The path π^*_{opt} is divided into $\mathfrak{u} = 5$ subpaths, each of which visits an (roughly) equal number of points. (ii) Since $\mathcal{E}_2 \geq \sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i/\mathfrak{u}$, we obtain the desired path π' by connecting p_{α_2} to p_{α_3}.

Proof: Set $\mathfrak{u} = \lceil 2/\varepsilon \rceil$. Let $\pi^*_{\text{opt}}(i,j) = \langle p_i, p_{i+1}, \ldots, p_j \rangle$ denote the portion of the path π^*_{opt} from p_i to p_j, and let $\mathcal{E}(i,j) = \left\| \pi^*_{\text{opt}}(i,j) \right\| - \|p_i - p_j\|$ denote its 2-excess. Let $\alpha_i = \lceil (i-1)(k-1)/\mathfrak{u} \rceil + 1$. By the definition, we have $\alpha_1 = 1$ and $\alpha_{\mathfrak{u}+1} = k$, and furthermore, each subpath $\pi^*_{\text{opt}}(\alpha_i, \alpha_{i+1})$ visits

$$\alpha_{i+1} - \alpha_i - 1 = \left(\left\lceil \frac{i(k-1)}{\mathfrak{u}} \right\rceil + 1 \right) - \left(\left\lceil \frac{(i-1)(k-1)}{\mathfrak{u}} \right\rceil + 1 \right) - 1 \leq \left\lfloor \frac{k-1}{\mathfrak{u}} \right\rfloor \tag{6.2}$$

points (excluding the endpoints p_{α_i} and $p_{\alpha_{i+1}}$).

Consider the subpaths $\pi^*_{\text{opt}}(\alpha_1, \alpha_2), \ldots, \pi^*_{\text{opt}}(\alpha_{\mathfrak{u}}, \alpha_{\mathfrak{u}+1})$ of π^*_{opt} and their 2-excesses $\mathcal{E}_1 = \mathcal{E}(\alpha_1, \alpha_2), \ldots, \mathcal{E}_{\mathfrak{u}} = \mathcal{E}(\alpha_{\mathfrak{u}}, \alpha_{\mathfrak{u}+1})$, respectively. Clearly, there exists an index ν, $1 \leq \nu \leq \mathfrak{u}$, such that $\mathcal{E}_\nu \geq (\sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i)/\mathfrak{u}$.

By connecting the vertex p_{α_ν} directly to the vertex $p_{\alpha_{\nu+1}}$ in π^*_{opt}, we obtain a new path $\pi' = \langle p_1, p_2, \ldots p_{\alpha_\nu}, p_{\alpha_{\nu+1}}, p_{\alpha_{\nu+1}+1}, \ldots, p_k \rangle$. Observe that $\|\pi'\| = \left\| \pi^*_{\text{opt}} \right\| - \mathcal{E}_\nu$, and by Eq. (6.2), π' visits at least $k - (\alpha_{\nu+1} - \alpha_\nu - 1) \geq k - \lfloor (k-1)/\mathfrak{u} \rfloor \geq (1 - 1/\mathfrak{u})k$ points of P. See Figure 6.6.

Consider the $(\mathfrak{u}+1)$-skeleton $\mathcal{S}' = \langle p_{\alpha_1}, p_{\alpha_2}, \ldots, p_{\alpha_{\mathfrak{u}+1}} \rangle$ of π'. By the definition of $\mathcal{E}_i$, we have that $\|\mathcal{S}'\| = \left\| \pi^*_{\text{opt}} \right\| - \sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i$. Therefore, by the definition of $\mathcal{E}_{\pi',\mathfrak{u}+1}$, we have that

$$\mathcal{E}_{\pi',\mathfrak{u}+1} \leq \|\pi'\| - \|\mathcal{S}'\| = \left(\left\| \pi^*_{\text{opt}} \right\| - \mathcal{E}_\nu \right) - \left(\left\| \pi^*_{\text{opt}} \right\| - \sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i \right) = \sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i - \mathcal{E}_\nu.$$

By applying Theorem 6.3.6 to the path π', one can compute a path ξ that visits $(1 - 1/\mathfrak{u})k \geq (1-\varepsilon)k$ points of P, of length

$$\begin{aligned} \|\xi\| &\leq \|\pi'\| + \frac{\mathcal{E}_{\pi',\mathfrak{u}+1}}{\mathfrak{u}+1} \leq \left(\left\| \pi^*_{\text{opt}} \right\| - \mathcal{E}_\nu \right) + \frac{1}{\mathfrak{u}+1} \left(\sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i - \mathcal{E}_\nu \right) \\ &= \left\| \pi^*_{\text{opt}} \right\| + \frac{1}{\mathfrak{u}+1} \left(\sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i - (\mathfrak{u}+2)\mathcal{E}_\nu \right) \leq \left\| \pi^*_{\text{opt}} \right\| \leq \mathcal{B}, \end{aligned}$$

since $\sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i - (\mathfrak{u}+2)\mathcal{E}_\nu \leq 0$, implied by $\mathcal{E}_\nu \geq (\sum_{i=1}^{\mathfrak{u}} \mathcal{E}_i)/\mathfrak{u}$. ∎

Of course, the value of k is not known in advance. Therefore, the algorithm tries all possible values of k from 1 to n, and returns the maximum value such that k points of P can be visited within the budget $\mathcal{B}$.

Theorem 6.4.2 *Given a set P of n points in $\mathbb{R}^d$, a budget $\mathcal{B}$, and a root r, let k_{opt} be the number of points of P visited by an optimal orienteering path starting at r with budget $\mathcal{B}$. One can compute, in $n^{O(d\sqrt{d}/\varepsilon)} \cdot (\sqrt{d}\log n/\varepsilon)^{(d/\varepsilon)^{O(d)}}$ time, a path that starts at r and visits at least $(1-\varepsilon)k_{\mathrm{opt}}$ points of P, and its length is at most $\mathcal{B}$.*

6.5 Conclusions

In this chapter, we defined the notion of $(\varepsilon, \mathfrak{u})$-approximation to k-TSP, and showed that Mitchell's k-TSP algorithm [Mit99] works actually as an $(\varepsilon, \mathfrak{u})$-approximation algorithm for the k-TSP problem in the plane. We used it to develop a $(1-\varepsilon)$-approximation algorithm for the orienteering problem. The analysis easily extends to handle the case where both the starting and ending vertex of the orienteering problem are specified. In particular, the algorithm can approximate the best orienteering cycle rooted at a point r.

Our algorithm sheds a light on the power of Mitchell's approach [Mit99] which has the advantage that it introduces errors only when the underlying path is "dense". This is in contrast to the Arora's technique [Aro98] which inherently introduces error in the approximation generated.

In the new analysis of the k-TSP algorithm the notion of surplus emerges naturally. We expect it to be much smaller than the excess in a lot of cases, and it might be of independent interest and useful in analyzing other algorithms.

There are numerous problems for further research, including:

- Can the running time be significantly improved?
- Can one extend the algorithms presented here to the problem of visiting points with time windows constraints [BES05, BBCM04, CP05], where one has to visit a point inside a prespecified time window? This problem seems to be more challenging. Currently, even a constant-factor approximation algorithm is not known for the simple case of visiting points on the line.

Part III

Conflict-free Coloring

Chapter 7

Introduction

A *range space* $(X, \mathcal{R})$ is defined by a ground set X and a family $\mathcal{R}$ of subsets of X, which are called *ranges* (for example, $X = \mathbb{R}^2$ and $\mathcal{R}$ is the set of all disks in the plane). A coloring of a set $P \subseteq X$ is *conflict-free* (CF for short) for $\mathcal{R}$ if for any range $R \in \mathcal{R}$ with $P \cap R \neq \emptyset$, there is at least one point in $P \cap R$ that has a unique color among the points of $P \cap R$. Namely, for any range $R \in \mathcal{R}$, there is a color that appears exactly once in the set $P \cap R$.

The problem of CF coloring is motivated by frequency assignment in wireless networks. Specifically, the points of P are *base stations* (or *antennas*) with a fixed transmission radius r, and the ranges are disks of radius r, centered at the *clients*. The colors are frequencies assigned to the antennas, and the conflict-free property means that any client can always find a frequency that is assigned to a unique antenna, among those that it can reach. In this case the communication with that antenna is free from interference with other antennas that are assigned the same frequency. The goal is then to minimize the number of distinct frequencies assigned to the antennas, while maintaining the conflict-free property.

The problem was introduced by Even *et al.* [ELRS03]. They showed that one can find an assignment of $O(\log n)$ frequencies to the base stations which is conflict-free for disks in the plane, and this is tight in the worst case. Har-Peled and Smorodinsky [HS05] extended those results by considering other range spaces. They gave sufficient conditions for the CF chromatic number to be small for more general ranges. The dual version of the CF coloring problem was studied in [ELRS03, HS05], where one colors the ranges so that, for any point, the set of ranges that contain the point is conflict-free. Smorodinsky [Smo06] improved several results studied in [ELRS03] by providing a deterministic coloring algorithms for those problems. For more variations on the online CF coloring problem see [BCS06].

The problem has been extended to the dynamic settings, in which the points of P (the base stations) are inserted one by one, starting with an empty set [FLM$^+$05, CFK$^+$07]. When a point is inserted, a color is assigned to it and the color cannot be changed later. The coloring should remain conflict-free at all times. Fiat *et al.* [FLM$^+$05] considered the case where P is a set of n points on the line, and $\mathcal{R}$ is the set of all intervals on the line. They present both deterministic and randomized algorithms for the problem. The best deterministic algorithm uses $O(\log^2 n)$ colors, and the best randomized algorithm uses $O(\log n \log\log n)$ colors with high probability. The best known lower bound for both randomized and deterministic algorithms, which also holds in the static case, is $\Omega(\log n)$ colors [PT03, Smo03].

The paper of Fiat *et al.* [FLM$^+$05] contains one negative result concerning online CF coloring of points in the plane for arbitrary disks as ranges. It shows that in the worst case n colors are needed (by any coloring algorithm). That is, there are situations where each newly inserted point requires a new color. (Recall, in contrast, that $O(\log n)$ colors suffice for the static case.)

7.1 Our results

In Chapter 9, we present randomized algorithms for online conflict-free coloring of points in the plane, with respect to intervals, halfplanes, congruent disks, and nearly-equal axis-parallel rectangles. In all four cases, the coloring algorithms use $O(\log n)$ colors, with high probability. We also present a deterministic algorithm for the CF coloring of points in the plane with respect to nearly-equal axis-parallel rectangles, using $O(\log^{12} n)$ colors. This is the first deterministic online CF coloring algorithm for this problem using only polylog(n) colors.

We start in Section 9.1 by presenting a randomized online algorithm for CF coloring of points on the line for intervals. We continue, in Section 9.2, with the related problem of online CF coloring of points in the plane for *halfplanes*. This is a simple generalization of the one-dimensional CF coloring problem. Indeed, if we restrict the two-dimensional problem to sets P of points on the upper unit semi-circle, and map the inserted points by projecting them on the x-axis, then the subsets of P that can be cut off the unit circle by halfplanes, when projected on the x-axis, are the same as the subsets of the projected set that can be cut off by intervals, or by complements of intervals.

The case of halfplanes is simpler than that of unit circles. However, it already demonstrates how geometry enters the analysis in a nontrivial manner. In Section 9.3, we extend this technique to the case of unit disks, using similar machinery. In particular, we obtain a randomized algorithm for online CF coloring of points in the plane for unit disks, that uses $O(\log n)$ colors with high probability. This is the main result in this chapter. Here, we also use the positive integers as the colors, and guarantee that the largest integer in each range is unique. The analysis of our algorithm, however, is more delicate. It is based on an observation that allows us, in certain cases, to charge a high color assigned to a point, to the disappearance of previously inserted points from the boundary of the convex hull of an appropriate subset of the (high-colored) points inserted so far. This charging scheme implies that the expected number of points that require color at least j decreases exponentially in j, thereby implying the logarithmic bound on the number of colors required.

In Section 9.4, we extend the approach to the problem of online CF coloring of points in the plane for nearly equal axis-parallel rectangles, namely, rectangles for which the ratio between the largest and the smallest widths, and the ratio between the largest and the smallest heights, are both bounded by some constant. Here too we obtain a coloring that uses, with high probability, $O(\log n)$ colors.

Notice that the offline version of all the problems we consider are quite easy. Offline CF coloring of n points for any of the three kinds of ranges mentioned above can be done with $O(\log n)$ colors, using, for example, the approach of [HS05] (see also Chapter 8). We recall that the known lower bound to the above problems, which also holds in the static cases, is $\Omega(\log n)$ colors [ELRS03, PT03].

Finally, in Section 9.5, we present a deterministic online algorithm for CF coloring with respect to nearly-equal axis-parallel rectangles in the plane. The algorithm uses $O(\log^{12} n)$ colors. This is the first deterministic online CF coloring algorithm for this problem that uses polylog n colors.

Computational model. When analyzing randomized online algorithms, there is a distinction between the *oblivious adversary model* and the *adaptive adversary* model. The oblivious adversary must construct the entire input sequence in advance, while the adaptive adversary may choose each input point based on the actions of the online algorithm made so far. We refer the interested reader to [BE98] for a discussion of these models. The analysis of all our algorithms is in the (weaker) oblivious adversary model. There are no known efficient randomized online algorithms for CF coloring against an adaptive adversary. In fact, it is an open question of whether one can bound the number of colors used by any of our algorithms when the adversary is adaptive.

The results appeared in [Che06a, CFK+07].

Chapter 8

Preliminaries

In this chapter, we briefly review the offline algorithm for the conflict-free coloring problems by Even *et al.* [ELRS03] and by Har-Peled and Smorodinsky [HS05]. In the following, P denotes a set of n points in $\mathbb{R}^d$, and $\mathcal{R}$ denotes a set of ranges (for example, the set of all discs in the plane).

Definition 8.0.1 The "Delaunay" graph $G = G(P, \mathcal{R})$ is the graph whose vertex set is P and whose edges are all pairs (u, v) for which there exists a range $r \in \mathcal{R}$ such that $r \cap P = \{u, v\}$.

A range space $(P, \mathcal{R})$ is *monotone* if for any $P_1 \subset P$ and for each $r \in \mathcal{R}$ with $|r \cap P_1| > 2$ there exists a range $r' \in \mathcal{R}$ such that $|r' \cap P_1| = 2$, and $r' \cap P_1 \subset r \cap P_1$.

A natural approach (used in [ELRS03]) for conflict-free coloring of a monotone range-space $(P, \mathcal{R})$, is to pick a large independent set L_1 in $G(P, \mathcal{R})$, color all the points of L_1 by a single color, and repeat this process on $(P \setminus L_1, \mathcal{R})$. This approach is summarized in Figure 8.1.

Let $L_i \subset P$ denote the set of points in P colored with i by CFColorAlg. We refer to L_i as the *ith layer* of $(P, \mathcal{R})$.

Lemma 8.0.2 ([ELRS03, HS05]) *The coloring of a monotone range space* $(P, \mathcal{R})$ *by* CFColorAlg *is a valid CF-coloring of* $(P, \mathcal{R})$*.*

Proof: Consider a range $r \in \mathcal{R}$, such that $|P \cap r| \geq 2$. Let i be the maximal color assigned to points of P lying in r. Let $P_i \subset P$ be the set of input points at the beginning of the ith iteration, i.e., the set just before color i has been assigned. Note that $L_i \subset P_i$ and $L_i \cap r = P_i \cap r$ (since i is the maximal color in r). Clearly, if $|r \cap L_i| = 1$ then r is served and we are done.

Thus, we only have to consider the case $|r \cap L_i| > 1$. However, by the monotonicity property (applied to the subset P_i), it follows that there exists a range r' such that: (i) $|r' \cap P_i| = 2$, and (ii) $r' \cap P_i \subset r \cap P_i = r \cap L_i$.

```
Algorithm CFColorAlg(P, R)
  i ← 0
  P_1 ← P
  while P_{i+1} ≠ 0 do
      i ← i + 1
      Find an independent set P'_i ⊆ P_i of G(P_i, R)
      Color: ∀x ∈ P'_i, f(x) ← i
      Prune: P_{i+1} ← P_i \ P'_i
```

Figure 8.1: An offline algorithm for CF coloring points in the plane.

This means that the two points of $r' \cap L_i$ form an edge in the graph $G(P_i, \mathcal{R})$. This however contradicts the fact that L_i is independent in $G(P_i, \mathcal{R})$. ■

Lemma 8.0.3 ([HS05]) *Let $\mathcal{R}$ be a set of ranges in $\mathbb{R}^d$, so that for any finite set P, the range space $(P, \mathcal{R})$ is monotone. If the Delaunay graph $G(P, \mathcal{R})$ contains an independent set of size at least $\alpha|P|$, for any finite set P and some fixed $0 < \alpha < 1$, then* CFColorAlg *uses at most $\frac{\log n}{\log(1/(1-\alpha))}$ colors for a set of n points.*

Proof: The assumption of the lemma implies that in the ith iteration of CFColorAlg we assign color i to at least $\alpha|P_i|$ points of P_i. It follows that if we start with a set of n points, the number of iterations is at most $\frac{\log n}{\log(1/(1-\alpha))}$. ■

Chapter 9

Online Conflict-free Coloring

In this chapter, we present randomized algorithms for online conflict-free coloring of points in the plane, with respect to intervals (on the line), halfplanes, congruent disks, and nearly-equal axis-parallel rectangles. In all these cases, the coloring algorithms use $O(\log n)$ colors, with high probability. We also present a deterministic algorithm for the CF coloring of points in the plane with respect to nearly-equal axis-parallel rectangles, using $O(\log^{12} n)$ colors. This is the first deterministic online CF coloring algorithm for this problem using only polylog(n) colors.

9.1 CF coloring for intervals

To motivate our 2-dimensional algorithms, we first present the randomized algorithm for CF coloring of points on the line for interval ranges. We identify the colors with the integers, so that there is a total order on the set of colors. The coloring produced by the algorithm is such that the *maximum* color in any (nonempty) interval is unique.

Let p be the next point inserted. We say that p *sees* a point x (alternatively, p sees the color $c(x)$) if all the colors of points between p and x (exclusive) have color smaller than $c(x)$. We say that p is *eligible* for color m if p does not see m. To give p a color, we scan all colors in increasing order. For each color i, if p is not eligible for color i we continue to color $i+1$. Otherwise, if p is eligible for color i, we set $c(p) = i$ with probability $1/2$, and continue to color $i+1$ with probability $1/2$.

It is easy to prove, by induction on the insertion order, that the maximum color in any interval is unique at any stage. Indeed, consider an interval T at some stage which contains at least two points of maximum color i (among those of the current points in T). Let p be the last inserted point that lies in T and got color i. By definition, when p was inserted it saw color i (with T as a "witness" interval), and therefore was not eligible for this color, contradicting the assumption that it has color i.

To show that this algorithm uses $O(\log n)$ colors with high probability, one argues that if the algorithm reaches color i when processing a point p, then p gets the color i with probability at least $1/8$. More formally, let C_i (resp., $C_{\geq i}$) be the (random variable) set of points of color i (resp., of color $\geq i$). Then

$$\mathbf{Pr}\Big[p \in C_i \mid p \in C_{\geq i}\Big] \geq \frac{1}{8} .$$

To see this, assume that p is neither the leftmost nor the rightmost point in $C_{\geq i}$ at the time of its insertion, and let q and r be its left and right neighbors in that set, respectively. In order for p to get color i, it is necessary that both q and r "advance" to higher colors, and that p "stays" at color i. The first two events happen together with probability at least $1/4$, and the conditional probability of the third event, conditioned on the first two occurring (and on p reaching $C_{\geq i}$), is $1/2$, since p does not see color i.[1] Hence, the probability of p to be in C_i, assuming it is in $C_{\geq i}$, is at least $1/8$, as claimed (the argument is simpler, and the probability is larger, when p is the leftmost or rightmost point in $C_{\geq i}$).

This implies that

$$\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq \frac{7}{8}\,\mathbf{E}\Big[|C_{\geq i}|\Big] \ .$$

Since $|C_{\geq 1}| = n$, we have, for $i \geq 1$,

$$\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq \left(\frac{7}{8}\right)^i n \ .$$

For $i = c\log_{8/7} n$, we get that $\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq 1/n^{c-1}$. Hence, by Markov's inequality,

$$\mathbf{Pr}\Big[|C_{\geq i+1}| \geq 1\Big] \leq 1/n^{c-1} \ ,$$

which shows that, with high probability, the algorithm uses only $i = O(\log n)$ colors.

9.2 CF coloring for halfplanes

In this section, we present an algorithm for CF coloring of points in the plane for halfplane ranges. The algorithm is similar to the one-dimensional algorithm of Section 9.1 but with a different definition of when a point p sees a color m. To simplify the presentation, we assume that the points of P are in *general position*, namely that no three of them are collinear.

Let p be the next point to be inserted. We say that p *sees* a point x (alternatively, p sees the color $c(x)$) if there is a halfplane h that contains x and p and no point of color higher than $c(x)$. As we will shortly argue, the coloring algorithm guarantees that in this case x is the only point of color $c(x)$ in h. We say that p is *eligible* for color m if p does not see m. To give p a color, we scan all colors in increasing order. For each color i, if p is not eligible for color i we continue to color $i+1$. Otherwise, if p is eligible for color i, we set $c(p) = i$ with probability $1/2$, and continue to color $i+1$ with probability $1/2$.

It is easy to prove by induction that the maximum color in any halfplane is unique at any stage. Indeed, consider a halfplane h at some stage which contains at least two points of maximum color i (among those of the current points in h). Let p be the last inserted point that lies in h and got color i. By definition, when p was inserted it saw color i (with h as a “witness” halfplane), and therefore was not eligible for this color, contradicting the assumption that it has color i. This also shows that if a newly inserted point p sees color i, then any halfplane that contains p, some points of color i, and no point of a larger color, must contain exactly one point of color i.

We next show that the algorithm uses $O(\log n)$ colors with high probability. Let C_i (resp., $C_{\geq i}$) be the set of points of color i (resp., of color $\geq i$). Let $B_{\geq i} \subseteq C_{\geq i}$ be the set of those points $p \in C_{\geq i}$ that see at least four other points of $C_{\geq i}$ when they are inserted. Let $E_{\geq i} = C_{\geq i} \setminus B_{\geq i}$. All these sets are random variables, depending on the random choices made by the algorithm.

Lemma 9.2.1 *Any point $p \in B_{\geq i}$ must lie outside the convex hull of the points in $C_{\geq i}$ that were inserted before it.*

[1] Note that this analysis strongly uses the fact that the adversary is oblivious.

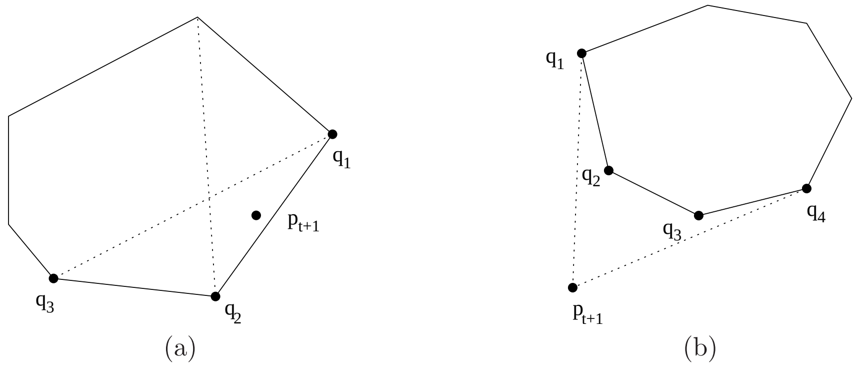


Figure 9.1: A point $p \in C_{\geq i}$ and the convex hull of the points in $C_{\geq i}$ inserted before p. (a) If p is inside the hull then it can see at most 3 points of $C_{\geq i}$. (b) p is outside the hull and it sees q_1, q_2, q_3, and q_4. Thus p is in $B_{\geq i}$.

Proof: Let A be the set of points of $C_{\geq i}$ inserted before p, and let $\mathcal{CH}(A)$ denote the convex hull of A. Assume to the contrary that $p \in B_{\geq i}$ and $p \in \mathcal{CH}(A)$. A point in $\mathcal{CH}(A)$ can only see vertices of $\mathcal{CH}(A)$ so if $\mathcal{CH}(A)$ has at most 3 vertices then $p \notin B_{\geq i}$, a contradiction.

So assume that $|\mathcal{CH}(A)| > 3$. Let $q_1, \ldots, q_h$ be the vertices of $\mathcal{CH}(A)$ in clockwise order. Assume that p sees q_2, say. Then p must be inside triangle $\triangle q_1 q_2 q_3$, because otherwise any halfplane that contains q_2 and p must contain q_1 or q_3 (or both), contradicting the assumption that p sees q_2 (we use here the property that the maximum color in any witness halfplane is unique).

Since p is within $\triangle q_1 q_2 q_3$, any halfplane that contains p must contain at least one point of q_1, q_2 and q_3. This implies that p cannot see any point other than q_1, q_2 and q_3, contradicting the assumption that $p \in B_{\geq i}$. See Figure 9.1. ■

Let $f = |C_{\geq i}|$ and let $p_1, \ldots, p_f$ be the points in $C_{\geq i}$ in the order in which they were inserted. For $1 \leq j \leq f$ let $A_j = \mathcal{CH}(\{p_1, \ldots, p_j\})$ (the convex hull of $\{p_1, \ldots, p_j\}$). By Lemma 9.2.1 and its proof, if $p_j \in B_{\geq i}$ then $A_j \neq A_{j-1}$. The point p_j is a vertex of A_j and all the at least four points that p sees when it is inserted are consecutive vertices of A_{j-1}. All these vertices except the first and the last are not vertices of A_j, and, since the hulls keep growing, nor are they vertices of any A_ℓ, for $\ell > j$. Thus each point $p_j \in B_{\geq i}$ removes at least two vertices from $\mathcal{CH}(A_{j-1})$, and no point of P is removed more than once. See Figure 9.1. This implies that $|B_{\geq i}| \leq \frac{1}{2}|C_{\geq i}|$ and thus $|E_{\geq i}| \geq \frac{1}{2}|C_{\geq i}|$.

Lemma 9.2.2

$$\mathbf{Pr}\Big[p \in C_i \mid p \in E_{\geq i}\Big] \geq \frac{1}{16}\ .$$

Proof: Fix the set $C_{\geq i}$ and consider only the coin flips that assign colors to the points of $C_{\geq i}$, after the points did reach $C_{\geq i}$ (note that, once $C_{\geq i}$ is fixed, the subsets $B_{\geq i}$ and $E_{\geq i}$ are also determined). Assume that $p \in E_{\geq i}$. By definition, the probability that p gets color i is 1/2 the probability that p is eligible for color i.

Point p is eligible for color i if all the points of $C_{\geq i}$ that it sees when it is inserted did not get color i. Since p sees at most three points of $C_{\geq i}$, the probability that none of them got color i is at least 1/8. ■

We thus obtain the following theorem.

Theorem 9.2.3 *The CF coloring algorithm of points for halfplanes presented in this section uses $O(\log n)$ colors with high probability.*

Proof: Using the same notation as above, since $|E_i| \geq |C_i|/2$, and since by Lemma 9.2.2 a point in E_i gets color i with probability $\geq 1/16$, we obtain that

$$\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq \left(1 - \frac{1}{32}\right) \mathbf{E}\Big[|C_{\geq i}|\Big] \ .$$

Since $|C_{\geq 1}| = n$, we have, for $i \geq 1$,

$$\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq \left(\frac{31}{32}\right)^i n \ .$$

For $i = c \log_{32/31} n$, we get that $\mathbf{E}\Big[|C_{\geq i+1}|\Big] \leq 1/n^{c-1}$. Hence, by Markov's inequality,

$$\mathbf{Pr}\Big[|C_{\geq i+1}| \geq 1\Big] \leq 1/n^{c-1} \ ,$$

from which the theorem follows. ■

9.3 CF coloring for congruent disks

We next extend the analysis of the preceding section to the case where the ranges are congruent disks of common radius 1, say. We tile the plane with axis-parallel squares of side 1/2, and assign to each of them a color class, so that no unit disk intersects two distinct squares with the same color class, and so that the total number of classes is a constant. Within each square we color the points independently, using the colors of the class assigned to the square.

Let Q be a square in the tiling. The coloring procedure for points in Q is identical to the one given for halfplanes, except that we say that p *sees* a point x if there is a unit disk D that contains x and p and no point of color higher than $c(x)$. As before, in this case, x is the only point of $D \cap Q$ of color $c(x)$. We say that p is *eligible* for color m if p does not see m, and apply the algorithm of Section 9.2 to the points in Q.

Correctness follows by induction, as in the preceding section, showing that for any unit disk D that contains points from a square Q, the maximum color of the points of $Q \cap D$ is unique.

We next bound the number of colors used by the algorithm. For any unit disk D that intersects Q, the center of D lies in an axis-parallel square Q_0 that is concentric with Q and has side length 5/2. Partition Q_0 into four disjoint equal sub-squares, $Q_0^1, \ldots, Q_0^4$, each an axis-parallel square of side length 5/4, and all having the center of Q as a common vertex. See Figure 9.2. Let $o^1, \ldots, o^4$ be the centers of $Q_0^1, \ldots, Q_0^4$, respectively. It is easy to check that a unit disk centered at o^d contains Q_0^d, for $d = 1, \ldots, 4$. This implies that each unit disk which intersects Q, contains at least one of the points $o^1, \ldots, o^4$. We arbitrarily associate each such unit disk with one of the points among $o^1, \ldots, o^4$ that it contains. We denote by $\mathcal{D}^d$ the set of unit disks associated with o^d. The following is a crucial property of this partitioning.

Lemma 9.3.1 *Let K^d denote the convex cone with apex o^d spanned by Q, for $d = 1, \ldots, 4$. Then, for any pair of disks $D, D' \in \mathcal{D}^d$, the intersection $\partial D \cap \partial D' \cap K^d$ consists of at most one point.*

Proof: Note that the opening angle of each of the cones K^d is smaller than $\pi/2$. Assume to the contrary that $\partial D \cap \partial D' \cap K^d$ contains two points, say x and y. Then $D \cap D' \cap K^d$ contains the triangle xo^dy,

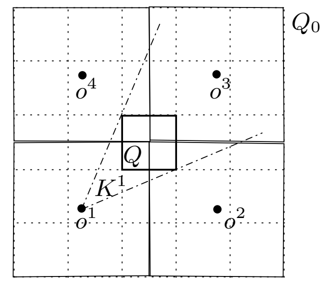


Figure 9.2: The partition of Q_0 into four sub-squares and the corresponding stabbing points o^d. The cone K^1 with apex o^1 spanned by Q is also shown.

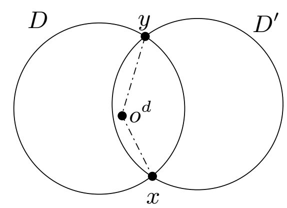


Figure 9.3: If $o^d \in D \cap D'$, the angle $\angle xo^dy$ has to be obtuse.

which is easily seen to imply that the angle $\angle xo^dy$ is greater than $\pi/2$; see Figure 9.3. This however is impossible, since this angle is smaller than the opening angle of K^d. ■

Let C_i (resp., $C_{\geq i}$) be the random variable which is equal to the set of points of color i (resp., of color $\geq i$). Let $B_{\geq i} \subseteq C_{\geq i}$ be the random variable that consists of any point $p \in C_{\geq i}$ that sees more than 36 other points of $C_{\geq i}$ when it is inserted. Let $E_{\geq i} = C_{\geq i} \setminus B_{\geq i}$.

In section 9.2 we controlled the sizes of the analogous sets $B_{\geq i}$, $E_{\geq i}$ by arguing that when a point of $B_{\geq i}$ is inserted, it removes at least two points from being vertices of the convex hull of $C_{\geq i}$ from this point on. To extend the argument to the case of unit disks, we replace the notion of convex hull vertices by *d-maximal vertices*, defined as follows.

Let I be a set of points in Q. For $d = 1, 2, \ldots, 4$, we define a point $p \in I$ to be *d-maximal* if there is a disk in $\mathcal{D}^d$ that contains p and no other point of I. Let $M^d(I)$ denote the subset of the d-maximal points in I.

Lemma 9.3.2 *Let $f = |C_{\geq i}|$ and let $p_1, \ldots p_f$ be the points in $C_{\geq i}$ in the order in which they are inserted. Let $A^d_j = M^d(\{p_1, \ldots, p_j\})$ for $d = 1, 2, 3, 4$. If $p_j \in B_{\geq i}$ then for some $d = 1, 2, 3, 4$, $|A^d_{j-1} \setminus A^d_j| \geq 8$. Moreover, the points of $A^d_{j-1} \setminus A^d_j$ will never again become d-maximal.*

Proof: Since $p_j \in B_{\geq i}$, p_j sees at least 37 points p_ℓ, $\ell < j$. That is, for each such point p_ℓ, there exists a unit disk D_ℓ containing only p_j and p_ℓ, among all points $p_1, \ldots, p_j$. For $d = 1, 2, 3, 4$, let H^d denote the subset of points p_ℓ for which the disk D_ℓ is in $\mathcal{D}^d$. Clearly, for at least one $d \in \{1, 2, 3, 4\}$, $|H^d| \geq 10$. Without loss of generality, assume that $|H^1| \geq 10$. It also follows by definition that the points in H^1 are 1-maximal in $\{p_1, \ldots, p_{j-1}\}$.

Let $m := |H^1|$ and let us also denote the points in H^1 by $q_1, \ldots, q_m$. Let $D_i \in \mathcal{D}^1$ be the unit disk

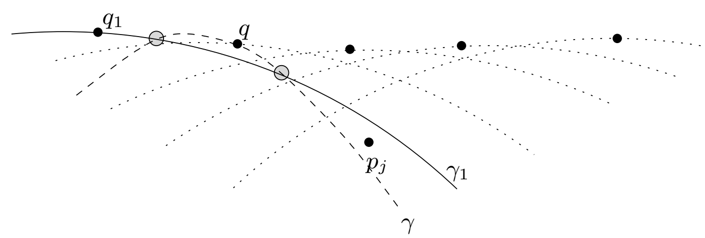


Figure 9.4: Illustrating the proof of Lemma 9.3.2. After inserting p_j every maximal point that it sees, except for the first and last in the θ-order, stops being maximal.

that contains p_j and q_i, and let γ_i denote the circle bounding D_i, for $i = 1, \ldots, m$.

Consider the situation in polar coordinates about the center $o = o^d$. Let $\theta_1 < \theta_2$ be the orientations of the two rays bounding the cone K^d defined in Lemma 9.3.1. We regard each γ_i as the graph of a function $\rho = \gamma_i(\theta)$, for $\theta_1 \leq \theta \leq \theta_2$. By Lemma 9.3.1, these graphs form a collection of θ-monotone *pseudolines*. By construction, p_j lies below (in the ρ-direction) all the graphs γ_i. Furthermore we can choose the disks D_i so that each point q_i lies on γ_i and above all the other graphs γ_j. That is, p_j lies below the lower envelope of the γ_i's, and the points q_i lie on the upper envelope of these arcs.

Without loss of generality, assume that the clockwise order (about o^d) of the points q_i along the envelope is $q_1, \ldots, q_m$. Let r be the index for which the θ-coordinate of p lies between those of q_r and q_{r+1}. We claim that all the points of H^1, except possibly for q_1 and q_m, are not in A^1_j.

Suppose, contrary to what the claim asserts, that there exists a unit disk $D \in \mathcal{D}^1$ that contains $q = q_\ell$ for some $1 < \ell < m$ but does not contain any other point of $\{p_1, \ldots, p_j\}$. Assume also that $\ell \leq r$, and let γ be the boundary of D.

The arc γ then passes below q_1, above q, and below p_j. On the other hand, the arc γ_1 passes above q_1, below q, and above p_j. Since q_1, q, p_j appear in this clockwise order about o^d, γ and γ_1 must intersect twice in K^1, contradicting the pseudoline property of these arcs. See Figure 9.4. The case where $j \geq r+1$ is treated similarly, with q_m playing the role of q_1. ■

Each point in $C_{\geq i}$ can leave A^d_j at most once, for each $d = 1, 2, 3, 4$. Therefore Lemma 9.3.2 implies that $|B_{\geq i}| \leq 4|C_{\geq i}|/8 = |C_{\geq i}|/2$. From here on, the proof continues exactly as in Section 9.2, leading to the following theorem.

Theorem 9.3.3 *The CF coloring algorithm of points for congruent disks presented in this section uses $O(\log n)$ colors with high probability.*

9.4 CF coloring for nearly equal axis-parallel rectangles

A (possibly infinite) family $\mathcal{F}$ of axis-parallel rectangles is a *family of nearly-equal* axis-parallel rectangles, if there exists some positive constant α, such that the ratio between the largest width and the smallest width of the rectangles of $\mathcal{F}$, and the ratio between the largest height and smallest height of the rectangles of $\mathcal{F}$, are both at most α.

Consider a family $\mathcal{F}$ of nearly-equal axis-parallel rectangles. By scaling the coordinate axes, we may assume that the width and the height of any rectangle in $\mathcal{F}$ lie in $[1, \alpha]$. We tile the plane with an axis-parallel square grid whose cells have side length 1/2. This ensures that both the width and the height of any rectangle in $\mathcal{F}$ are larger than the side length of a square tile of the grid. We assign to each grid tile a color class, so that no rectangle in $\mathcal{F}$ intersects two distinct grid tiles with the same color class. As in the case of unit disks, it is easy to verify that a constant number (indeed, $O(\alpha^2)$) of color classes suffices. We assign colors to points within each grid tile independently, using the colors of the class assigned to the tile. Let Q be an arbitrary square tile. By the discussion above, we can assume (without loss of generality) that all the points are inserted into the interior of Q.

The algorithm for online CF coloring of the points within Q is the same as the algorithm of Section 9.3. Here, an incoming point p *sees* a point x if there is a rectangle in $\mathcal{F}$ that contains x and p and no point of color higher than $c(x)$.

The analysis is analogous to the analysis of Section 9.3. Here the corners of Q, denoted by o^1, o^2, o^3, and o^4, play the same role in the analysis as o^1, o^2, o^3, and o^4 in the previous section. That is, each rectangle in $\mathcal{F}$ that intersects Q contains at least one corner of Q, as is easily checked, and we arbitrarily associate it with one of the corners that it contains. Let $\mathcal{F}^i$ be the set of rectangles associated with o^i, for $i = 1, \ldots, 4$. The rest of the proof is similar to the one in Section 9.3, and is based on the easy observation that the boundaries of the rectangles in each fixed subfamily $\mathcal{F}^i$ behave as pseudolines within Q. Hence we have

Theorem 9.4.1 *The coloring algorithm always produces a conflict-free coloring, and the number of colors that it uses is $O(\log n)$, with high probability.*

Remark: If the rectangles in $\mathcal{F}$ are not nearly equal then, even in the static case, the number of colors required by the best known CF coloring algorithm is close to $\sqrt{n}$ [HS05] (see also [AKS99, PT03]). The intuitive reason that we can extend our approach and improve this bound for nearly-equal rectangles is the fact that if R and R' are two nearly equal rectangles whose boundaries intersect, then any pair of boundary intersection points lie "far apart" from each other, unless R and R' slightly overlap each other near a vertex of each (and this latter case is bypassed by the analysis, as then R and R' are placed in different subfamilies $\mathcal{F}^i$). In contrast, two nearly equal disks can almost overlap one another and yet the two intersections of their boundaries can be arbitrarily close to each other.

In other words, for our algorithm to work, it is crucial that the boundaries of the ranges behave like *pseudolines*. For halfplanes this holds trivially, whereas for congruent disks and nearly equal axis-parallel rectangles the property is enforced by tiling the plane, focussing on a single tile, and partitioning the ranges into subfamilies.

9.5 Deterministic CF coloring for nearly-equal axis-parallel rectangles

In this section, we present a *deterministic* online algorithm for online CF coloring a sequence P of points in the plane for a family $\mathcal{F}$ of nearly equal axis-parallel rectangles, which uses $O(\log^{12} n)$ colors. As discussed in Section 9.4, we can assume that the points of P all lie in a fixed square Q, whose side length is smaller than the width and the height of any rectangle of $\mathcal{F}$.

Definition 9.5.1 A *quadrant* is an unbounded region of the plane whose boundary is formed by two rays,

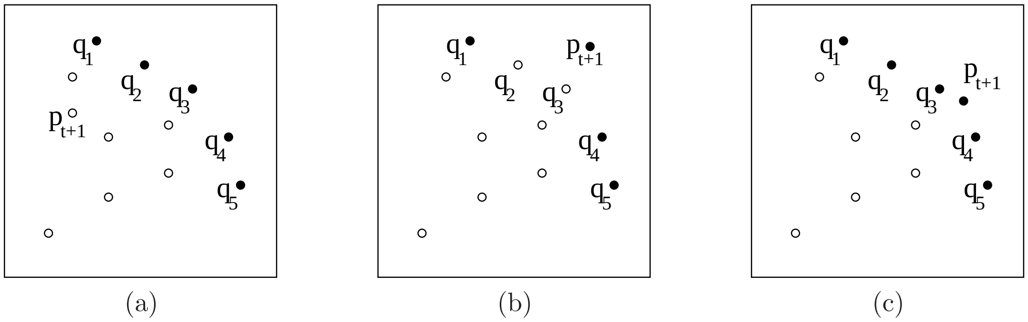


Figure 9.5: Illustrating the hull $\mathcal{H}(P(t+1))$ after p_{t+1} is inserted. The points of $\mathcal{H}(P(t+1))$ are black. In (a), p_{t+1} is dominated by q_1, q_2, q_3. In (b), p_{t+1} dominates q_2, q_3. In (c), p_{t+1} neither dominates nor is dominated by q_1, q_2, q_3, q_4, or q_5.

one parallel to the x-axis and the other parallel to the y-axis. The common source point of these two rays is the *corner* of the quadrant. A quadrant D is a *right-top quadrant* if $D = \{c(D)+(x, y) \mid x, y \geq 0\}$, where $c(D)$ denotes the corner of D. *Left-top*, *right-bottom*, and *left-bottom* quadrants are defined similarly.

We need the following simple decomposition lemma.

Lemma 9.5.2 *Let $(X, \mathcal{R})$ be a range space that can be decomposed into β range spaces $(X, \mathcal{R}_j)$, for $j = 1, \ldots, \beta$, where $\mathcal{R} = \cup_j \mathcal{R}_j$. Assuming that algorithm Alg_j can CF color a sequence of n points with respect to $(X, \mathcal{R}_j)$ using $f_j(n)$ colors, then one can CF color a sequence of n points with respect to $(X, \mathcal{R})$ using $\prod_j f_j(n)$ colors.*

Proof: We assign an incoming point p the color $A(p) = \langle A_1(p), \ldots, A_\beta(p) \rangle$, where $A_j(p)$ denotes the color assigned to p by the jth algorithm Alg_j.

The correctness is immediate. Indeed, consider a range $\mathbf{r}$ and the point set $P(t)$ after t insertions. Suppose $\mathbf{r} \in \mathcal{R}_k$; then there exists a point $q \in \mathbf{r} \cap P(t)$ such that q has the unique color $A_k(q)$ among the colors assigned (by Alg_k) to the points of $\mathbf{r} \cap P(t)$, by the correctness of Alg_k. Clearly, $A(q) \neq A(p)$ for all $p \in \mathbf{r} \cap P(t)$ and $p \neq q$. ■

We remind the reader that every rectangle of $\mathcal{F}$ is larger than a fixed square Q; as such every rectangle of $\mathcal{F}$ intersects Q like a quadrant of the following four types of quadrants: right-top, left-top, left-bottom, and right-bottom. Therefore, Lemma 9.5.2 implies that it is sufficient to show how to CF color the points with respect to, say, right-top quadrants. Indeed, by using the same algorithm independently four times (with appropriate reflections of the plane) to color the points with respect to the four different types of quadrants, we get a new algorithm that can CF color for rectangles of $\mathcal{F}$.

9.5.1 Preliminaries

We need an algorithm for the following problem DynCFProb of coloring points on the line: At each time, we either (i) insert a new point onto the line, or (ii) replace all the points in an interval with a new point; and we wish to color the points online using positive integers, so that there exists a unique highest colored point in any interval at all times.

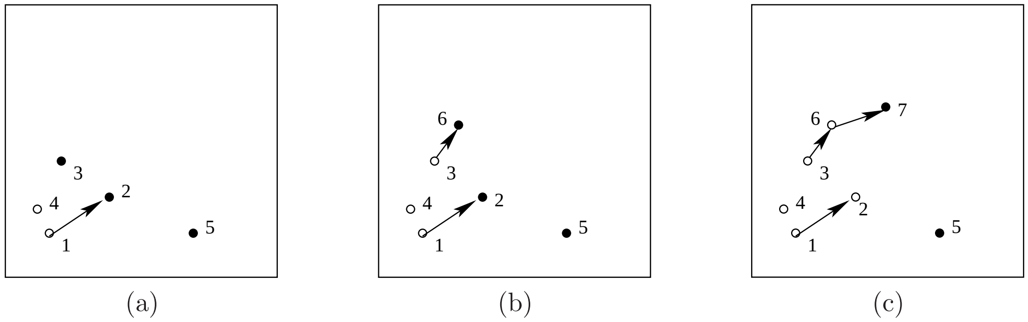


Figure 9.6: Illustrating the directed paths of the points of P. The numbers beside the points show the order they are inserted. The points on the hull are black. In (a), the R-values of p_1, p_2, p_3, p_4, p_5 are $1, 1, 2, 0, 3$, respectively. In (b), p_6 dominates p_3 and is assigned the R-value of p_3, which is 2. In (c), p_7 dominates p_2 and p_6, and is assigned the higher of the R-values of p_2 and p_6, which is 2.

Fiat *et al.* [FLM$^+$05] presented a deterministic algorithm for the following related problem, using $O(\log^2 n)$ colors: At each time, we insert a new point onto the line, and we wish to color the points online using positive integers, so that there exists a unique highest colored point in any interval at all times. Note that the algorithm of Fiat *et al.* [FLM$^+$05] can be immediately adapted to solving DynCFProb: For the insertion operation, just apply the algorithm of Fiat *et al.*; for the replacement operation, supposing that p_{t+1} is replacing all the points of interval I, then we assign the highest color in I to p_{t+1}. (Note that the replacement operation is only performed conceptually, by "collapsing" all the points of I into p_{t+1}.) We refer to this adapted algorithm as DynCFAlg.

We also need an algorithm for the following problem SuffixCFProb of coloring points on the line: At each time, we insert a new point onto the line to the right of last point inserted; and we wish to color the points using positive integers, so that there exists a unique highest colored point for each suffix at all times. There is a simple algorithm for this problem: For $i \geq 1$, let $b(i)$ be the position of the rightmost '1' bit in the binary representation of i. (For example, $b(1), \ldots, b(10)$ is $1, 2, 1, 3, 1, 2, 1, 4, 1, 2$, respectively.) If p is the ith point inserted onto the line then we assign $b(i)$ to p. We refer to this algorithm as SuffixCFAlg. It is easy to verify the following claim.

Claim 9.5.3 SuffixCFAlg *uses at most* $\lfloor \log_2 n \rfloor + 1$ *colors for* n *points, such that for each suffix of the list of the points on the line, there exists a unique point of the highest color.*

9.5.2 Deterministic online CF coloring for right-top quadrants

For simplicity of exposition, we assume that the points of P have distinct x and y coordinates. We would like to color P with respect to the range space $(\mathbb{R}^2, \mathcal{F}^{\mathrm{RT}})$, where $\mathcal{F}^{\mathrm{RT}}$ is the set of all right-top quadrants. We use the RGB color metaphor to describe colors of points. As such, a point is assigned a R-value and a G-value, which together as a pair form the color of the point.

Definition 9.5.4 A point p *dominates* a point q if $p >_x q$ and $p >_y q$; namely, any right-top quadrant that contains q must also contain p.

A point $p \in P(t)$ is *maximal in* $P(t)$ if no point in $P(t)$ dominates p. The *hull* of $P(t)$ is the sorted list of the maximal points of $P(t)$ in increasing x-coordinate order, and is denoted by $\mathcal{H}(P(t))$.

Observation 9.5.5 *Let $\mathcal{H}(P(t)) = \langle q_1, \ldots, q_k \rangle$, where $q_1 <_x \ldots <_x q_k$. The following holds:*

(i) If a point p dominates some points of $\mathcal{H}(P(t))$, then it must dominate a consecutive block of points of $\mathcal{H}(P(t))$. That is, if q_b and q_c are dominated by p, then $q_b, \ldots, q_c$ are dominated by p.

(ii) If a quadrant $D \in \mathcal{F}^{RT}$ contains some points of $\mathcal{H}(P(t))$, then $D \cap \mathcal{H}(P(t))$ consists of a consecutive block of points of $\mathcal{H}(P(t))$.

Suppose that $\mathcal{H}(P(t)) = \langle q_1, \ldots, q_k \rangle$. When the point p_{t+1} is inserted, there are three possibilities. (i) If p_{t+1} is dominated by a point of $\mathcal{H}(P(t))$, then $\mathcal{H}(P(t+1))$ remains the same as $\mathcal{H}(P(t))$. (ii) If p_{t+1} dominates a consecutive block of points of $\mathcal{H}(P(t))$, say $q_b, \ldots, q_c$, then $\mathcal{H}(P(t+1)) = \langle q_1, \ldots, q_{b-1}, p_{t+1}, q_{c+1}, \ldots, q_k \rangle$. (iii) Otherwise, there must exist an index b such that $q_b <_x p_{t+1} <_x q_{b+1}$, and we have $\mathcal{H}(P(t+1)) = \langle q_1, \ldots, q_b, p_{t+1}, q_{b+1}, \ldots, q_k \rangle$. See Figure 9.5.

Observe that one can view the hull as an instance of DynCFProb: When inserting p_{t+1}, case (i) above does not change the hull; case (ii) corresponds to a replacement operation in DynCFProb; and case (iii) corresponds to an insertion operation in DynCFProb. Therefore, when inserting p_{t+1}, in case (i), we assign 0 to be the *R-value* of p_{t+1}; and in cases (ii) and (iii), we use DynCFAlg to assign a positive integer (with respect to the hull) to be the *R-value* of p_{t+1}. The points with R-values equal to zero are *trivial points*.

This following claim is implied directly from the correctness of DynCFAlg.

Claim 9.5.6 *At any time t, for a consecutive block of points of the hull $\mathcal{H}(P(t))$, there exists a unique highest R-value in the block.*

In particular, in case (ii) above, in the block of points on $\mathcal{H}(P(t))$ dominated by p_{t+1}, there is an unique point, say q_i, that realizes the highest R-value (also note that p_{t+1} "inherits" the R-value from q_i). We add a directed edge from q_i to p_{t+1}. Therefore, we create a number of *directed paths* of the points of P. For a point p, the target of the directed path to which p belongs is the *leader* of p, and is denoted by leader(p). (If a directed path has only a single point p then the leader of p is itself.) It is easy to verify that the R-values of the points on a directed path are all the same. See Figure 9.6.

Lemma 9.5.7 *Consider a quadrant $D \in \mathcal{F}^{RT}$ and let i be the highest R-value assigned to a point of $D \cap P(t)$. Then all the points with R-value i in $D \cap P(t)$ form a suffix of a directed path. Furthermore, the leader of the points of this directed path is on the hull $\mathcal{H}(P(t))$.*

Proof: Let U denote the set of points in $D \cap P(t)$ with R-value equal to i. We first show that for any $p \in U$, leader(p) must be on the hull. Suppose for the contrary that q is the leader of p and q is not on the hull. Since q is not on the hull, it must have been dominated by some point, say, q', in $D \cap P(t)$. Since q is the leader of p, there is no directed edge from q to q', which implies that q' "inherited" a R-value higher than q has. But this contradicts the assumption that q has the highest R-value i in $D \cap P(t)$.

Note that for any $p \in U$, all the points after p in the directed path to which p belongs must also be in U. This implies that U is formed by a union of suffices of directed paths.

Next, assume, for the sake of contradiction, that $p, q \in U$, and p is not an ancestor of q and q is not an ancestor of p. Then leader(p) and leader(q) must be distinct points on the hull. But this is impossible since it would imply that the interval encompassing leader(p) and leader(q) on the hull has the maximal R-value (i.e., i) appearing twice (which contradicts the correctness of DynCFAlg). Thus, U must be formed by a suffix of a single directed path. ■

Having assigned the R-values to the points, the algorithm still fails to provide us with a valid CF coloring of the points, since several points on a directed path (having the same R-values) are still conflicting with each other. However, when inserting a new (non-trivial) point to the hull, either we add this point to the end of a specific directed path, or alternatively, it is the first vertex in a new directed path. Thus, for a new (non-trivial) point, we assign it a G-value according to its position in its directed path. In particular, we assign a point a G-value by using SuffixCFAlg on the directed path the point is being added to. The color of a point p is the pair (R_p, G_p), where R_p and G_p are the R-value and G-value assigned to p, respectively.

Lemma 9.5.8 *The generated coloring is conflict-free with respect to right-top quadrants.*

Proof: Consider any quadrant $D \in \mathcal{F}^{RT}$ and let i be the maximal R-value in $D \in D^{RT}$. By Lemma 9.5.7, all the points that have R-value i in D form a suffix of a directed path π. We know that there is a unique G-value assigned to one of those points on π by Claim 9.5.3. Thus, the point on π that realizes this G-value has a unique color in D. ■

Lemma 9.5.9 *For a sequence of n points, the algorithm uses $O(\log^3 n)$ colors.*

Proof: The algorithm assigns $O(\log^2 n)$ distinct R-values, because DynCFAlg uses this number of colors for a sequence of n points. To complete the proof, observe that the algorithm assigns at most $\lfloor \log_2 n \rfloor + 1$ different G-values, by Claim 9.5.3. ■

9.5.3 The result

Combining Lemma 9.5.2, Lemma 9.5.8 and Lemma 9.5.9 together implies the following result.

Theorem 9.5.10 *One can* deterministically *online color a sequence of n points in the plane, such that the coloring is always conflict-free with respect to a family of nearly-equal axis-parallel rectangles. The algorithm uses $O\big(\log^{12} n\big)$ colors.*

This result has been improved by Chen *et al.* [CFK+07]. They show that $O(\log^3 n)$ colors suffice for this problem.

9.6 Conclusions

In this chapter, we presented randomized online CF coloring algorithms (against oblivious adversaries) for several range space in the plane, using $O(\log n)$ colors with high probability. We also presented the first *deterministic* algorithm for CF coloring points in the plane with respect to nearly-equal axis-parallel rectangles (which works against a non-oblivious adversary) that uses $\operatorname{polylog} n$ colors.

Interestingly, we were unable to extend the deterministic algorithm to other ranges (in particular, halfplanes, and congruent disks) in the plane, and we leave as open the problem of finding any deterministic algorithm for these ranges that uses only polylogarithmically many colors. Another open problem is to obtain randomized algorithms with comparable performances against non-oblivious adversaries.

References

[AGK$^+$01] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 21–29, 2001.

[AGK$^+$04] 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.

[AH98] E. Arkin and R. Hassin. On local search for weighted k-set packing. *Math. of Oper. Res.*, 23:640–648, 1998.

[AHV04] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[AK03] S. Arora and G. Karakostas. Approximation schemes for minimum latency problems. *SIAM J. Comput.*, 32(5):1317–1337, 2003.

[AKS99] N. Alon, M. Krivelevich, and B. Sudakov. Coloring graphs with sparse neighborhoods. *J. Combinat. Theory, Ser. B*, 77:73–82, 1999.

[AMN98] E. M. Arkin, J. S. B. Mitchell, and G. Narasimhan. Resource-constrained geometric network optimization. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 307–316, 1998.

[AR06] A. Aboud and Y. Rabani. Correlation clustering with penalties. manuscript, 2006.

[Aro98] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. *J. Assoc. Comput. Mach.*, 45(5):753–782, 1998.

[Aro03] S. Arora. Approximation schemes for NP-hard geometric optimization problems: a survey. *Math. Prog.*, 97:43–69, 2003.

[BBCM04] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 166–174, 2004.

[BC03] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. In *Proc. 14th ACM-SIAM Sympos. Discrete Algo.*, pages 801–802, 2003.

[BCK$^+$03] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. In *Proc. 44th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 46–55, 2003.

[BCR01] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 11–20, 2001.

[BCS06] A. Bar-Noy, P. Cheilaris, and S. Smorodinsky. Conflict-free coloring for intervals: from offline to online. In *Proc. 18th Annu. ACM Sympos. Parallelism Algo. Architectures, (SPAA'06)*, pages 128–137, 2006.

[BE98] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, New York, NY, USA, 1998.

[BES05] R. Bar-Yehuda, G. Even, and S. Shahar. On approximating a geometric prize-collecting traveling salesman problem with time windows. *J. Algorithms*, 55(1):76–92, 2005.

[BS80] J. L. Bentley and J. B. Saxe. Decomposable searching problems I: Static-to-dynamic transformation. *J. Algo.*, 1(4):301–358, 1980.

[CFK+07] K. Chen, A. Fiat, H. Kaplan, M. Levy, J. Matoušek, E. Mossel, J. Pach, M. Sharir, S. Smorodinsky, U. Wagner, and E. Welzl. Online conflict-free coloring for intervals. *SIAM J. Comput.*, 36(5):1342–1359, 2007.

[CG99] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 378–388, 1999.

[CGTS02] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.

[CH06] K. Chen and S. Har-Peled. The orienteering problem in the plane revisited. In *Proc. 22nd Annu. ACM Sympos. Comput. Geom.*, pages 247–254, 2006.

[CH07] K. Chen and S. Har-Peled. The Euclidean orienteering problem revisited. *SIAM J. Comput.*, 2007. In press.

[Che06a] K. Chen. How to play a coloring game against a color-blind adversary. In *Proc. 22nd Annu. ACM Sympos. Comput. Geom.*, pages 44–51, 2006.

[Che06b] K. Chen. On k-median clustering in high dimensions. In *Proc. 17th ACM-SIAM Sympos. Discrete Algo.*, pages 1177–1185, 2006.

[Che08] K. Chen. A constant factor approximation algorithm for k-median clustering with outliers. In *Proc. 19th ACM-SIAM Sympos. Discrete Algo.*, 2008. To appear.

[CKMN01] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. 12th ACM-SIAM Sympos. Discrete Algo.*, pages 642–651, 2001.

[COP03] M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 30–39, 2003.

[CP05] C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *Proc. 46th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 245–253, 2005.

[CR05] J. Chuzhoy and Y. Rabani. Approximating k-median with non-uniform capacities. In *Proc. 16th ACM-SIAM Sympos. Discrete Algo.*, pages 952–958, 2005.

[ELRS03] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM J. Comput.*, 33(1):94–136, 2003.

[ES04] M. Effros and L. J. Schulman. Deterministic clustering with data nets. Technical Report TR04-050, Elec. Colloq. Comp. Complexity, 2004.

[FLM+05] A. Fiat, M. Levy, J. Matoušek, E. Mossel, J. Pach, M. Sharir, S. Smorodinsky, U. Wagner, and E. Welzl. Online conflict-free coloring for intervals. In *Proc. 16th ACM-SIAM Sympos. Discrete Algo.*, pages 545–554, 2005. Full version appeared in SICOMP.

[FS05] G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *Proc. 37th Annu. ACM Sympos. Theory Comput.*, pages 209–217, 2005.

[Gar05] N. Garg. Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In *Proc. 37th Annu. ACM Sympos. Theory Comput.*, pages 396–402, 2005.

[GMMO00] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *Proc. 41th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 359–366, 2000.

[Goe06] M. X. Goemans. Minimum bounded degree spanning trees. In *Proc. 47nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 273–282, 2006.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[HK05] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. In *Proc. 21st Annu. ACM Sympos. Comput. Geom.*, pages 126–134, 2005.

[HM04a] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[HM04b] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[HS05] S. Har-Peled and S. Smorodinsky. Conflict-free coloring of points and simple regions in the plane. *Discrete Comput. Geom.*, 34(1):47–70, 2005.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 604–613, 1998.

[Ind99] P. Indyk. Sublinear time algorithms for metric space problems. In *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 154–159, 1999.

[Ind04] P. Indyk. Algorithms for dynamic geometric problems over data streams. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 373–380, 2004.

[JMM+03] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.

[JV99] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–13, 1999.

[JV01] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. Assoc. Comput. Mach.*, 48(2):274–296, 2001.

[KBP03] S. Khuller, R. Bhatia, and R. Pless. On local search and placement of meters in networks. *SIAM J. Comput.*, pages 470–487, 2003.

[Khu05] S. Khuller. Problems column. *ACM Trans. Algorithms*, 1(1):157–159, 2005.

[KPR00] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000.

[KR00] J. Konemann and R. Ravi. A matter of degree: improved approximation algorithms for degree-bounded minimum spanning trees. In *Proc. 32st Annu. ACM Sympos. Theory Comput.*, pages 537–546, 2000.

[KSS04] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proc. 45nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 454–462, 2004.

[KSS05] A. Kumar, Y. Sabharwal, and S. Sen. Linear time algorithms for clustering problems in any dimensions. In *Proc. 32nd Int. Colloq. Automata Lang. Prog.*, pages 1374–1385, 2005.

[Llo82] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Info. Theory*, 28(2):129–136, 1982.

[LV92] J-H Lin and J. S. Vitter. ε-approximations with minimum packing constraint violation (extended abstract). In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 771–782, 1992.

[Mah04] M. Mahdian. *Facility Location and the Analysis of Algorithms through Factor-Revealing Programs*. Ph.D. dissertation, MIT, Department of Computer Science, 2004.

[Mat00] J. Matoušek. On approximate geometric k-clustering. *Discrete Comput. Geom.*, 24:61–84, 2000.

[Mit99] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, 28:1298–1309, 1999.

[Mit00] J. S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rudiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, chapter 15, pages 633–701. Elsevier, 2000.

[MOP01] N. Mishra, D. Oblinger, and L. Pitt. Sublinear time approximate clustering. In *Proc. 12th ACM-SIAM Sympos. Discrete Algo.*, pages 439–447, 2001.

[MP04] R. R. Mettu and C. G. Plaxton. Optimal time bounds for approximate clustering. *Mach. Learn.*, 56(1-3):35–60, 2004.

[PT03] J. Pach and G. Toth. Conflict-free colorings. *Discrete Comput. Geom., The Goodman-Pollack Festschrift*, pages 665–671, 2003.

[RRPS04] D. Ren, I. Rahal, W. Perrizo, and K. Scott. A vertical distance-based outlier detection method with local pruning. In *Proc. 13th ACM Conf. Information and Knowledge Management*, pages 279–284, 2004.

[SL07] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. 39th Annu. ACM Sympos. Theory Comput.*, pages 661–670, 2007.

[Smo03] S. Smorodinsky. *Combinatorial Problems in Computational Geometry*. Ph.D. thesis, Tel-Aviv University, School of Computer Science, 2003.

[Smo06] S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *Proc. 17th ACM-SIAM Sympos. Discrete Algo.*, pages 316–323, 2006.

[ST06] Z. Svitkina and E. Tardos. Approximation algorithm for facility location with hierarchical facility costs. In *Proc. 17th ACM-SIAM Sympos. Discrete Algo.*, pages 1088–1097, 2006.

[Tho05] M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2005.

[TV02] P. Toth and D. Vigo, editors. *The vehicle routing problem.* SIAM Monographs on Discrete Mathematics and Applications, 2002.

Curriculum Vitæ

Ke Chen

Department of Computer Science
University of Illinois at Urbana-Champaign

Research Interests

Approximation algorithms, Clustering algorithms and their applications, geometric computing, data streaming algorithms, randomized algorithms

Education

2001–present **Ph.D.**, Computer Science, University of Illinois at Urbana-Champaign
Research supervised by Prof. Sariel Har-Peled

1999–2001 **M.Eng.**, Computer Science, Huazhong University of Science and Technology, Wuhan, China
Thesis title: A Neural Network Based Intrusion Detection System
Research supervised by Prof. Zhitang Li

1995–1999 **B.Eng.**, Computer Science, Huazhong University of Science and Technology, Wuhan, China
Graduated with honors

Employment

2007 **Research Assistant**, Dept. of Computer Science, University of Illinois at Urbana-Champaign
Worked with Prof. Sariel Har-Peled.

2005–2006 **Teaching Assistant**, Dept. of Computer Science, University of Illinois at Urbana-Champaign
Assists teaching the courses "Undergraduate algorithms" and "Introduction to theory of computation". The main responsibility includes leading problem-solving sessions, hosting office hours, and grading homeworks.

2002–2005 **Research Assistant**, Dept. of Computer Science, University of Illinois at Urbana-Champaign
Worked with Prof. Sariel Har-Peled. The research focused on the design and analysis of algorithms for clustering and other geometric problems.

2001–2002 **Research Assistant**, Dept. of Computer Science, University of Illinois at Urbana-Champaign
Worked with Prof. Daniel Reed. The research focused on automatic SCSI hard drive characterization.

Conference Presentations

2006 On k-median clustering in high dimensions.
17th ACM-SIAM Symposium on Discrete Algorithms, Miami, FL.

2006 How to play a coloring game against a color-blind adversary.
22nd Annual ACM Symposium on Computational Geometry, Sedona, AZ.

2006 The orienteering problem in the plane revisited.
22nd Annual ACM Symposium on Computational Geometry, Sedona, AZ.

Honors & Awards

- Departmental and University-wide scholarships, Huazhong University of Science and Technology, B.Eng. studies
- Scholarship for outstanding graduate students, Huazhong University of Science and Technology (2000)
- Engineering college travel award, University of Illinois at Urbana-Champaign (2006)

Publications

Journal Papers

[1] Ke Chen, Amos Fiat, Haim Kaplan, Meital Levy, Jiří Matoušek, Elchanan Mossel, János Pach, Micha Sharir, Shakhar Smorodinsky, Uli Wagner, and Emo Welzl. Online conflict-free coloring for intervals. In *SIAM Journal on Computing (SICOMP)*, 2006, vol. 36, 2006, pp. 1342-1359.

[2] Ke Chen and Sariel Har-Peled. The Euclidean orienteering problem revisited. To appear in *SIAM Journal on Computing (SICOMP)*.

Refereed Conferences

[3] Ke Chen. On k-median clustering in high dimensions. In *Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006, pp. 1177–1185.

[4] Ke Chen. How to play a coloring game against a color-blind adversary. In *Proceedings of 22nd Annual ACM Symposium on Computational Geometry (SoCG)*, 2006, pp. 44–51.

[5] Ke Chen and Sariel Har-Peled. The orienteering problem in the plane revisited. In *Proceedings of 22nd Annual ACM Symposium on Computational Geometry (SoCG)*, 2006, pp. 247–254.

[6] Ke Chen. A constant factor approximation algorithm for k-median clustering with outliers. To appear in *Proceedings of 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

Professional Service

Refereeing

Journals: SIAM Journal on Computing, International Journal of Computational Geometry and Applications.

Conferences: ACM Symposium on Theory of Computing, ACM-SIAM Symposium on Discrete Algorithms, European Symposium on Algorithms.