
Article:
Cohen-Addad, V., Feldmann, A.E. orcid.org/0000-0001-6229-5332 and Saulpic, D. orcid.org/0000-0003-4208-8541 (2021) Near-linear time approximation schemes for clustering in doubling metrics. Journal of the ACM, 68 (6). 44. ISSN 0004-5411

https://doi.org/10.1145/3477541

Near-Linear Time Approximation Schemes for Clustering in Doubling Metrics

Vincent Cohen-Addad[1], Andreas Emil Feldmann*[2], and David Saulpic[3]

[1]Sorbonne Université, UPMC Univ Paris 06, CNRS, LIP6, Paris, France (vcohenad@gmail.com)

[2]Charles University in Prague, Czechia (feldmann.a.e@gmail.com)

[3]École Normale Supérieure, Paris (david.saulpic@ens.fr)

Abstract

We consider the classic Facility Location, k-Median, and k-Means problems in metric spaces of doubling dimension d. We give nearly linear-time approximation schemes for each problem. The complexity of our algorithms is $\widetilde{O}(2^{(1/\varepsilon)^{O(d^2)}}n)$, making a significant improvement over the state-of-the-art algorithms which run in time $n^{(d/\varepsilon)^{O(d)}}$.

Moreover, we show how to extend the techniques used to get the first efficient approximation schemes for the problems of prize-collecting k-Median and k-Means, and efficient bicriteria approximation schemes for k-Median with outliers, k-Means with outliers and k-Center.

*Supported by the Czech Science Foundation GAČR (grant #19-27871X), and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/SCI/004).

1 Introduction

The k-Median and k-Means problems are classic clustering problems that are highly popular for modeling the problem of computing a "good" partition of a set of points of a metric space into k parts so that points that are "close" should be in the same part. Since a good clustering of a dataset allows to retrieve information from the underlying data, the k-Median and k-Means problems are the cornerstone of various approaches in data analysis and machine learning. The design of efficient algorithms for these clustering problems has thus become an important challenge.

The input for the problems is a set of points in a metric space and the objective is to identify a set of k centers C such that the sum of the pth power of the distance from each point of the metric to its closest center in C is minimized. In the k-Median problem, p is set to 1 while in the k-Means problem, p is set to 2. In general metric spaces both problems are known to be APX-hard, and this hardness even extends to Euclidean spaces of any dimension $d = \Omega(\log n)$ [5]. Both problems also remain NP-hard for points in $\mathbb{R}^2$ [33]. For k-Center, the goal is to minimize the maximum distance from each point in the metric to its closest center. This problem is APX-hard even in Euclidean Spaces [17], and computing a solution with optimal cost but $(1+\varepsilon)k$ centers requires time at least $\Omega(n^{\sqrt{1/\varepsilon}})$ [30]. Therefore, to get an efficient approximation scheme one needs to approximate both the number of centers and the cost. (See Section 1.3 for more related work).

To bypass these hardness of approximation results, researchers have considered low-dimensional inputs like Euclidean spaces of fixed dimension or more generally metrics of fixed doubling dimension. There has been a large body of work to design good tools for clustering in metrics of fixed doubling dimension, from the general result of Talwar [37] to very recent coreset constructions for clustering problems [23]. In their seminal work, Arora et al. [4] gave a polynomial time approximation scheme (PTAS) for k-Median in $\mathbb{R}^2$, which generalizes to a quasi-polynomial time approximation scheme (QPTAS) for inputs in $\mathbb{R}^d$. This result was improved in two ways. First by Talwar [37] who generalized the result to any metric space of fixed doubling dimension. Second by Kolliopoulos and Rao [25] who obtained an $f(\varepsilon, d) \cdot n \log^{d+6} n$ time algorithm for k-Median in d-dimensional Euclidean space. Unfortunately, Kolliopoulos and Rao's algorithm relies on the Euclidean structure of the input and does not immediately generalize to low dimensional doubling metric. Thus, until recently the only result known for k-Median in metrics of fixed doubling dimension was a QPTAS. This was also the case for slightly simpler problems such as Uniform Facility Location. Moreover, as pointed out in [12], the classic approach of Arora et al. [4] cannot work for the k-Means problem. Thus no efficient algorithms were known for the k-Means problem, even in the plane.

Recently, Friggstad et al. [19] and Cohen-Addad et al. [13] showed that the classic local search algorithm for the problems gives a $(1+\varepsilon)$-approximation in time $n^{1/\varepsilon^{O(d)}}$ in Euclidean space, in time $n^{O(1/\varepsilon^2)}$ for planar graphs (which also extends to minor-free graphs), and in

time $n^{(d/\varepsilon)^{O(d)}}$ in metrics of doubling dimension d [19]. More recently Cohen-Addad [12] showed how to speed up the local search algorithm for Euclidean space to obtain a PTAS with running time $nk(\log n)^{(d/\varepsilon)^{O(d)}}$.

Nonetheless, obtaining an efficient approximation scheme (namely an algorithm running in time $f(\varepsilon, d)\text{poly}(n)$) for k-Median and k-Means in metrics of doubling dimension d has remained a major challenge.

The versatility of the techniques we develop to tackle these problems allows us to consider a broader setting, where the clients do not necessarily have to be served. In the prize-collecting version of the problems, every client has a penalty cost that can be paid instead of its serving cost. In the k-Median (resp. k-Means) with outliers problems, the goal is to serve all but z clients, and the cost is measured on the remaining ones with the k-Median (resp. k-Means) cost. These objectives can help to handle some noise from the input: the k-Median objective can be dramatically perturbed by the addition of a few distant clients, which must then be discarded.

1.1 Our Results

We solve this open problem by proposing the first near-linear time algorithms for the k-Median and k-Means problems in metrics of fixed doubling dimension. More precisely, we show the following theorems.

Theorem 1.1. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon)$-approximation algorithm for k-Median in metrics of doubling dimension d with running time $2^{(1/\varepsilon)^{O(d^2)}} n \log^4 n + 2^{O(d)} n \log^9 n$ and success probability at least $1 - 2\varepsilon$.*

Theorem 1.2. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon)$-approximation algorithm for k-Means in metrics of doubling dimension d with running time $2^{(1/\varepsilon)^{O(d^2)}} n \log^5 n + 2^{O(d)} n \log^9 n$ and success probability at least $1 - O(\varepsilon)$.*

Our results also extend to the Facility Location problem, in which no bound on the number of opened centers is given, but each center comes with an opening cost. The aim is to minimize the sum of the (1st power) of the distances from each point of the metric to its closest center, in addition to the total opening costs of all used centers.

Theorem 1.3. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon)$-approximation algorithm for Facility Location in metrics of doubling dimension d with running time $2^{(1/\varepsilon)^{O(d^2)}} n + 2^{O(d)} n \log n$ and success probability at least $1 - \varepsilon$.*

In all these theorems, we make the common assumption to have access to the distances of the metric in constant time, as, e.g., in [22, 15, 20]. This assumption is discussed in Bartal et al. [7].

Note that the double-exponential dependence on d is unavoidable unless P = NP, since the problems are APX-hard in Euclidean space of dimension $d = O(\log n)$. For Euclidean inputs, our algorithms for the k-Means and k-Median problems outperform the ones of Cohen-Addad [12], removing in particular the dependence on k, and the one of Kolliopoulos and Rao [25] when $d > 3$, by removing the dependence on $\log^{d+6} n$. Interestingly, for $k = \omega(\log^9 n)$ our algorithm for the k-Means problem is faster than popular heuristics like k-Means++ which runs in time $O(nk)$ in Euclidean space.

We note that the success probability can be boosted to $1-\varepsilon^\delta$ by repeating the algorithm $\log \delta$ times and outputting the best solution encountered.

After proving the three theorems above, we will apply the techniques to prove the following ones. We say an algorithm is an (α, β)-approximation for k-Median or k-Means with outliers if its cost is within an α factor of the optimal one and the solution drops βz outliers. Similarly, an algorithm is an (α, β)-approximation for k-Center if its cost is within an α factor of the optimal one and the solution opens βk centers.

Theorem 1.4. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon)$-approximation algorithm for Prize-Collecting k-Median (resp. k-Means) in metrics of doubling dimension d with running time $2^{(1/\varepsilon)^{O(d^2)}} n \log^4 n + 2^{O(d)} n \log^9 n$ and success probability at least $1-\varepsilon$.*

Theorem 1.5. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon, 1+O(\varepsilon))$-approximation algorithm for k-Median (resp. k-Means) with outliers in metrics of doubling dimension d with running time $2^{(1/\varepsilon)^{O(d^2)}} n \log^6 n + T(n)$ and success probability at least $1-\varepsilon$, where $T(n)$ is the running time to construct a constant-factor approximation.*

We note as an aside that our proof of Theorem 1.5 could give an approximation where at most z outliers are dropped, but $(1+O(\varepsilon))k$ centers are opened. For simplicity, we focused on the previous case.

Theorem 1.6. *For any $0 < \varepsilon < 1/3$, there exists a randomized $(1+\varepsilon, 1+O(\varepsilon))$-approximation algorithm for k-Center in metrics of doubling dimension d, with running time $2^{(1/\varepsilon)^{O(d^2)}} n \log^6 n + n \log k$ and success probability at least $1-\varepsilon$.*

As explained above, this bicriteria is necessary in order to get an efficient algorithm: it is APX-hard to approximate the cost [17], and achieving the optimal cost with $(1+\varepsilon)k$ centers requires a complexity $\Omega(n^{1/\sqrt{\varepsilon}})$ [30]. To the best of our knowledge, this works presents the first linear-time bicriteria approximation scheme for the problem of k-center.

1.2 Techniques

To give a detailed insight on our techniques and our contribution we first need to quickly review previous approaches for obtaining approximation schemes on bounded doubling metrics. The general approach, due to Arora [3] and Mitchell [36], which was generalized to doubling metrics by Talwar [37], is the following.

1.2.1 Previous Techniques

The approach consists in randomly partitioning the metric into a constant number of regions, and applying this recursively to each region. The recursion stops when the regions contain only a constant number of input points. This leads to what is called a *split-tree decomposition*: a partition of the space into a finer and finer level of granularity. The reader who is not familiar with the split-tree decomposition may refer to Section 2.2 for a more formal introduction.

Portals The approach then identifies a specific set of points for each region, called *portals*, which allows to show that there exists a near-optimal solution such that different regions "interplay" only through portals. For example, in the case of the Traveling Salesperson (TSP) problem, it is possible to show that there exists a near-optimal tour that enters and leaves a region only through its portals. In the case of the k-Median problem a client located in a specific region can be assigned to a facility in a different region only through a path that goes to a portal of the region. In other words, clients can "leave" a region only through the portals.

Proving the existence of such a structured near-optimal solution relies on the fact that the probability that two very close points end up in different regions of large diameter is very unlikely. Hence the expected *detour* paid by going through a portal of the region is small compared to the original distance between the two points, if the portals are dense enough.

For the sake of argument, we provide a proof sketch of the standard proof of Arora [3]. We will use a refined version of this idea in later sections. The split-tree recursively divides the input metric (V, dist) into parts of smaller and smaller diameter. The root part consists of the entire point set and the parts at level i are of diameter roughly 2^i. The set of portals of a part of level i is an $\varepsilon_0 2^i$-*net* for some ε_0, which is a small set such that every point of the metric is at distance at most $\varepsilon_0 2^i$ to it. Consider two points u, v and let us bound the expected detour incurred by connecting u to v through portals. This detour is determined by a path that starting from u at the lowest level, in each step connects a vertex w to its closest net point of the part containing w on the next higher level. This is done until the lowest-level part $R_{u,v}$ (i.e., the part of smallest diameter) is reached, which contains both u and v, from where a similar procedure leads from this level through portals of smaller and smaller levels all the way down to v. If the level of $R_{u,v}$ is i then the detour, i.e., the difference between $\text{dist}(u, v)$ and the length of the path connecting u and v through portals, is $O(\varepsilon_0 2^i)$ by the definition of the net. Moreover, the proof shows that the probability that u and v are not in the same part on level i is at most $\text{dist}(u, v)/2^i$. Thus, the expected detour for connecting u to v is $\sum_{\text{level } i} \Pr[R_{u,v} \text{ is at level } i] \cdot O(\varepsilon_0 2^i) = \sum_{\text{level } i} O(\varepsilon_0 \text{dist}(u, v))$. Hence, setting ε_0 to be some ε divided by the number of levels yields that the expected detour is $O(\varepsilon \text{dist}(u, v))$.

Dynamic programming The portals now act as separators between different parts and allows to apply a dynamic programming (DP) approach for solving the problems. The DP consists of a DP-table entry for each part and for each *configuration* of the portals of the part. Here a configuration is a potential way the near-optimal solution interacts with the part. For example, in the case of TSP, a configuration is the information at which portal the near-optimal tour enters and leaves and how it connects the portals on the outside and inside of the part. For the k-Median problem, a configuration stores how many clients outside (respectively inside) the part connect through each portal and are served by a center located inside (respectively outside). Then the dynamic program proceeds in a bottom-up fashion along the split-tree to fill up the DP table. The running time of the dynamic program depends exponentially on the number of portals.

How many portals? The challenges that need to be overcome when applying this approach, and in particular to clustering problems, are two-fold. First the "standard" use of the split-tree requires $O((\frac{\log n}{\varepsilon})^d)$ portals per part in order to obtain a $(1+\varepsilon)$-approximation, coming from the fact that the number of levels can be assumed to be logarithmic in the number of input points. This often implies quasi-polynomial time approximation schemes since the running time of the dynamic program has exponential dependence on the number of portals. This is indeed the case in the original paper by Talwar [37] and in the first result on clustering in Euclidean space by Arora et al. [4]. However, in some cases, one can lower the number of portals per part needed. In Euclidean space for example, the celebrated "patching lemma" [2] shows that only a constant number (depending on ε) of portals are needed for TSP. Similarly, Kolliopoulos and Rao [25] showed that for k-Median in Euclidean space only a constant number of portal are needed, if one uses a slightly different decomposition of the metric.

Surprisingly, obtaining such a result for doubling metrics is much more challenging. To the best of our knowledge, this work is the first one to reduce the number of portals to a constant.

A second challenge when working with split-tree decompositions and the k-Means problem is that because the cost of assigning a point to a center is the squared distance, the analysis of Arora, Mitchell, and Talwar does not apply. If two points are separated at a high level of the split-tree, then making a detour to the closest portal may incur an expected cost much higher than the cost of the optimal solution.

1.2.2 Our Contributions

Our contribution can be viewed as a "patching lemma" for clustering problems in doubling metrics. Namely, an approach that allows to solve the problems mentioned above: (1) it shows how to reduce the number of portals to a constant, (2) it works for any clustering objective which is defined as the sum of distances to some constant p (with k-Median and

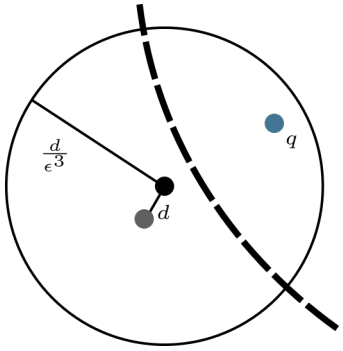

Figure 1: Illustration of badly cut. The black point is c (resp. l), the gray one is $L(c)$ (resp. f_0), and the blue point is q. The dashed line is the boundary of a part with "large" diameter.

k-Means as prominent special cases), and (3) it works not only for Euclidean but also for doubling metrics.

Our starting point is the notion of *badly cut* vertices of Cohen-Addad [11] for the capacitated version of the above clustering problems. To provide some intuition on the definition, let us focus on k-median and start with the following observation: consider a center f of the optimal solution and a client c assigned to f. If the diameter of the lowest-level part containing both f and c is of order $\text{dist}(c, f)$ (say at most $\text{dist}(c, f)/\varepsilon^2$), then by taking a large enough but constant size net as a set of portals in each part (say an $\varepsilon^3 2^i$-net for a part of level i), the total detour for the two points is at most $O(\varepsilon \text{dist}(c, f))$, which is acceptable.

The problematic scenario is when the lowest-level part containing f and c is of diameter much larger than $\text{dist}(c, f)$. In this case, it is impossible to afford a detour proportional to the diameter of the part in the case of the k-Median and k-Means objectives. To handle this case we first compute a constant approximation L (via some known algorithm) and use it to guide us towards a $(1 + \varepsilon)$-approximation.

Badly cut clients and facilities Consider a client c and the center $L(c)$ serving c in L (i.e., $L(c)$ is closest to c among the centers in L), and call $\text{OPT}(c)$ the facility of an optimum solution OPT that serves c in OPT. We say that c is *badly cut* if there is a point q in the ball centered at c of radius $\text{dist}(c, L(c))/\varepsilon$ such that the highest-level part containing c and not q is of diameter much larger than $\text{dist}(c, L(c))/\varepsilon$ (say greater than $\text{dist}(c, L(c))/\varepsilon^2$). In other words, there is a point q in this ball such that paying a detour through the portal to connect c to q yields a detour larger than $\varepsilon \text{dist}(c, q)$ (see Figure 1).

Similarly, we say that a center l is *badly cut* if there is a point q in the ball centered at l of radius $\text{dist}(l, f_0)/\varepsilon$ (where f_0 is the facility of OPT that is the closest to l) such that the highest-level part containing l and not q is of diameter $\text{dist}(l, f_0)/\varepsilon^2$. The crucial

property here is that any client c or any facility l is badly cut with probability $O(\varepsilon^3)$, as we will show.

Using the notion of badly cut We now illustrate how this notion can help us. Assume for simplicity that $\mathrm{OPT}(c)$ is in the ball centered at a client c of radius $\mathrm{dist}(c, L(c))/\varepsilon$ (if this is not the case then $\mathrm{dist}(c, \mathrm{OPT}(c))$ is much larger than $\mathrm{dist}(c, L(c))$, so this is a less problematic scenario and a simple idea can handle it). If c is not badly cut, then the lowest-level part containing both c and $\mathrm{OPT}(c)$ is of diameter not much larger than $\mathrm{dist}(c, L(c))/\varepsilon$. Taking a sufficiently fine net for each part (independent of the number of levels) allows to bound the detour through the portals to reach $\mathrm{OPT}(c)$ from c by at most $\varepsilon\mathrm{dist}(c, L(c))$. Since L is an $O(1)$-approximation, this is fine.

If c is badly cut, then we modify the instance by relocating c to $L(c)$. That is, we will work with the instance where there is no more client at c and there is an additional client at $L(c)$. We claim that any solution in the modified instance can be lifted to the original instance at an expected additional cost of $O(\varepsilon^3\mathrm{OPT})$. This comes from the fact that the cost increase for a solution is, by the triangle inequality, at most the sum of distances of the badly cut clients to their closest facility in the local solution. This is at most $O(\varepsilon^3\mathrm{OPT})$ in expectation since each client is badly cut with probability at most $O(\varepsilon^3)$ and L is an $O(1)$-approximation.

Here we should ask, what did we achieve by moving c to $L(c)$? Note that c should now be assigned to facility f of OPT that is the closest to $L(c)$. So we can make the following observation: If $L(c)$ is not badly cut, then the detour through the portals when assigning c to f is fine (namely at most ε times the distance from $L(c)$ to its closest facility in OPT). Otherwise, if $L(c)$ is also badly cut, then we simply argue that there exists a near-optimal solution which contains $L(c)$, in which case c is now served optimally at a cost of 0 (in the new instance).

From bicriteria to opening exactly k centers Since $L(c)$ is badly cut with probability $O(\varepsilon^3)$, this leads to a solution opening $(1 + O(\varepsilon^3))k$ centers. At first, it looks difficult to then reduce the number of centers to k without increasing the cost of the solution by a factor larger than $(1 + \varepsilon)$. However, and perhaps surprisingly, we show in Lemma 4.6 that this can be avoided: we show that there exists a near-optimal solution that contains the badly cut centers of $L(c)$.

We can then conclude that a near-optimal solution can be computed by a simple dynamic-programming procedure on the split-tree decomposition to identify the best solution in the modified instance.

Our result on Facility Location in Section 3 provides a simple illustration of these ideas — avoiding the bicriteria issue due to the hard bound on the number of opened facilities for the k-Median and k-Means problems. Our main result on k-Median and k-Means is described in Section 4. We discuss some extensions of the framework in Section 5.

1.3 Related work

On clustering problems The clustering problems considered in this paper are known to be NP-hard, even restricted to inputs lying in the Euclidean plane (see Mahajan et al. [29] or Dasgupta and Freund [16] for k-Means, Megiddo and Supowit [34] for the problems with outliers, and Masuyama et al. [31] for k-Center). The problems of Facility Location and k-Median have been studied since a long time in graphs, see e.g. [24]. The current best approximation ratio for metric Facility Location is 1.488, due to Li [27], whereas it is 2.67 for k-Median, due to Byrka et al. [8].

The problem of k-Means in general graphs also received a lot of attention (see e.g., Kanungo et al. [24]) and the best approximation ratio is 6.357, due to Ahmadian et al. [1].

Clustering problems with outliers where first studied by Charikar et al. [10], who devised an $(O(1), (1 + O(\varepsilon))$-approximation for k-Median with outliers and a constant factor approximation for prize-collecting k-Median. More recently, Friggstad et al. [18] showed that local search provides a bicriteria approximation, where the number of centers is approximate instead of the number of outliers. However, the runtime is $n^{f(\varepsilon,d)}$, and thus we provide a much faster algorithm. To the best of our knowledge, we present the first approximation scheme that preserves the number of centers.

The k-Center problem is known to be NP-hard to approximate within any factor better than 2, a bound that can be achieved by a greedy algorithm [17]. This is related to the problem of covering points with a minimum number of disks (see e.g. [28, 30]). Marx and Pilipczuk [30] proposed an exact algorithm running in time $n^{\sqrt{k}+O(1)}$ to find the maximum number of points covered by k disks and showed a matching lower bound, whereas Liao et al. [28] presented an algorithm running in time $O(mn^{O(1/\varepsilon^2 \log^2 1/\varepsilon)})$ to find a $(1 + \varepsilon)$-approximation to the minimal number of disks necessary to cover all the points (where m is the total number of disks and n the number of points). This problem is closely related to k-Center: the optimal value of k-Center on a set V is the minimal number L such that there exist k disks of radius L centered on points of V covering all points of V. Hence, the algorithm from [28] can be directly extended to find a solution to k-Center with $(1 + \varepsilon)k$ centers and optimal cost. The local search algorithm of Cohen-Addad et al. [] can be adapted to k-center and generalizes the last result to any dimension d: in $\mathbb{R}^d$, one can find a solution with optimal cost and $(1 + \varepsilon)k$ centers in time $n^{1/\epsilon^{O(d)}}$. Loosing on the approximation allows us to present a much faster algorithm.

On doubling dimension Despite their hardness in general metrics, these problems admit a PTAS when the input is restricted to a low dimensional metric space: Friggstad et al. [19] showed that local search gives a $(1 + \varepsilon)$-approximation. However, the running time of their algorithm is $n^{(d/\varepsilon)^{O(d)}}$ in metrics with doubling dimension d.

A long line of research exists on filling the gap between results for Euclidean spaces and metrics with bounded doubling dimension. This started with the work of Talwar [37], who gave QPTASs for a long list of problems. The complexity for some of these problems

was improved later on: for the Traveling Salesperson problem, Gottlieb [20] gave a near-linear time approximation scheme, Chan et al. [9] gave a PTAS for Steiner Forest, and Gottlieb [20] described an efficient spanner construction.

2 Preliminaries

2.1 Definitions

Consider a metric space (V, dist). For a vertex $v \in V$ and an integer $r \geq 0$, we let $\beta(v, r) = \{w \in V \mid \text{dist}(v, w) \leq r\}$ be the *ball* around v with radius r. The *doubling dimension* of a metric is the smallest integer d such that any ball of radius $2r$ can be covered by 2^d balls of radius r. We call Δ the aspect-ratio (sometimes referred to as *spread* in the literature) of the metric, i.e., the ratio between the largest and the smallest distance.

Given a set of points called *clients* and a set of points called *candidate centers* in a metric space, the goal of the *k-Median* problem is to output a set of k *centers* (or *facilities*) chosen among the candidate centers that minimizes the sum of the distances from each client to its closest center. More formally, an instance to the k-Median problem is a 4-tuple (C, F, dist, k), where $(C \cup F, \text{dist})$ is a metric space and k is a positive integer. The goal is to find a set $S \subseteq F$ such that $|S| \leq k$ and $\sum_{c \in C} \min_{f \in S}(\text{dist}(c, f))$ is minimized. Let $n = |C \cup F|$. The *k-Means* problem is identical except from the objective function which is $\sum_{c \in C} \min_{f \in S}(\text{dist}(c, f))^2$.

In the *Facility Location* problem, the number of centers in the solution is not limited but there is a cost w_f for each candidate center f and the goal is to find a solution S minimizing $\sum_{c \in C} \min_{f \in S}(\text{dist}(c, f)) + \sum_{f \in S} w_f$.

For those clustering problems, it is convenient to name the center serving a client. For a client c and a solution S, we denote $S(c)$ the center closest to c, and $S_c := \text{dist}(c, S(c))$ the distance to it.

In this paper, we consider the case where the set of candidate centers is part of the input. A variant of the k-Median and k-Means problems in Euclidean metrics allows to place centers anywhere in the space and specifies the input size as simply the number of clients. We note that up to losing a polylogarithmic factor in the running time, it is possible to reduce this variant to our setting by computing a set of candidate centers that approximate the best set of centers in $\mathbb{R}^d$ [32].

A *δ-net* of V is a set of points $X \subseteq V$ such that for all $v \in V$ there is an $x \in X$ such that $\text{dist}(v, x) \leq \delta$, and for all $x, y \in X$ we have $\text{dist}(x, y) > \delta$. A net is therefore a set of points not too close to each other, such that every point of the metric is close to a net point. The following lemma bounds the cardinality of a net in doubling metrics.

Lemma 2.1 (from Gupta et. al [21])**.** *Let (V, d) by a metric space with doubling dimension d and diameter D, and let X be a δ-net of V. Then $|X| \leq 2^{d \cdot \lceil \log_2(D/\delta) \rceil}$.*

Another property of doubling metrics that will be useful for our purpose is the existence of low-stretch spanners with a linear number of edges. More precisely, Har-Peled and Mendel [22] showed that one can find a graph (called a *spanner*) in the input metric that has $O(n)$ edges such that distances in the graph approximate the original distances up to a constant factor. This construction takes time $2^{O(d)}n$. We will make use of these spanners only for computing constant-factor approximations of our problems: for this purpose, we will therefore assume that the number of edges is $m = 2^{O(d)}n$.

We will also make use the following lemma.

Lemma 2.2 ([14]). *Let $p \geq 0$ and $1/2 > x > 0$. For any $a, b, c \in V$, we have* $\text{dist}(a,b)^p \leq (1+x)^p \text{dist}(a,c)^p + \text{dist}(c,b)^p(1+1/x)^p$.

2.2 Decomposition of Metric Spaces

As pointed out in our techniques section, we will make use of hierarchical decompositions of the input metric. We define a *hierarchical decomposition* (sometimes simply a decomposition) of a metric (V, dist) as a collection of partitions $\mathcal{D} = \{\mathcal{B}_0, \ldots, \mathcal{B}_{|\mathcal{D}|}\}$ that satisfies the following:

- each $\mathcal{B}_i$ is a partition of V,
- $\mathcal{B}_i$ is a refinement of $\mathcal{B}_{i+1}$, namely for each part $B \in \mathcal{B}_i$ there exists a part $B' \in \mathcal{B}_{i+1}$ that contains B,
- $\mathcal{B}_0$ contains a singleton set for each $v \in V$, while $\mathcal{B}_{|\mathcal{D}|}$ is a trivial partition that contains only one set, namely V.

We define the *ith level* of the decomposition to be the partition $\mathcal{B}_i$, and call $B \in \mathcal{B}_i$ a level-i part. If $B' \in \mathcal{B}_{i-1}$ is such that $B' \subset B$, we say that B' is a *subpart* of B.

For a given decomposition $\mathcal{D} = \{\mathcal{B}_0, \ldots, \mathcal{B}_{|\mathcal{D}|}\}$, we say that a vertex u is *cut from v at level j* if j is the maximum integer such that v is in some $B \in \mathcal{B}_j$ and u is in some $B' \in \mathcal{B}_j$ with $B \neq B'$. For a vertex v and radius r we say that the ball $\beta(v, r)$ is *cut* by $\mathcal{D}$ at level j if j is the maximum level for which some vertex of the ball is cut from v at level j.

A key ingredient for our result is the following lemma, that introduces some properties of the hierarchical decomposition (sometimes referred to as *split-tree*) proposed by Talwar [37] for low-doubling metrics.

Lemma 2.3 (Reformulation of [37, 6])**.** *For any metric (V, dist) of doubling dimension d and any $\rho > 0$, there is a randomized hierarchical decomposition $\mathcal{D}$ such that the diameter of a part $B \in \mathcal{B}_i$ is at most 2^{i+1}, $|\mathcal{D}| \leq \lceil \log_2(\mathrm{diam}(V)) \rceil$, each part $B \in \mathcal{B}_i$ is refined in at most $2^{O(d)}$ parts at level $i-1$, and:*

1. ***Scaling probability**: for any $v \in V$, radius r, and level i, we have*
$$\Pr[\mathcal{D} \text{ cuts } \beta(v, r) \text{ at a level } i] \leq 2^{2d+2} r / 2^i.$$
2. ***Portal set:** every set $B \in \mathcal{B}_i$ where $\mathcal{B}_i \in \mathcal{D}$ comes with a set of* portals $\mathcal{P}_B \subseteq B$ *that is*
   (a) ***concise:** the size of the portal set is bounded by $|\mathcal{P}_B| \leq 1/\rho^d$; and*
   (b) ***precise:** for every node $u \in B$ there is a portal $p \in \mathcal{P}_B$ close-by, i.e., $\mathrm{dist}(u, p) \leq \rho 2^{i+1}$; and*
   (c) ***nested:** any portal of level $i+1$ that lies in B is also a portal of B, i.e., for every $p \in \mathcal{P}_{B'} \cap B$ where $B' \in \mathcal{B}_{i+1}$ we have $p \in \mathcal{P}_B$.*

Moreover, this decomposition can be found in time $(1/\rho)^{O(d)} n \log \Delta$.

2.3 Formal Definition of Badly Cut Vertices

As sketched in the introduction, the notion of badly cut lies at the heart of our analysis. We define it formally here. We denote $\kappa(\varepsilon, p) = \varepsilon^2 \frac{p}{(p+\varepsilon)^p}$ and $\tau(\varepsilon, d) = 2d + 2 + \log(1/\kappa(\varepsilon, p))$, two parameters that are often used throughout this paper.

Definition 2.4. *Let $(C \cup F, \mathrm{dist})$ be a metric with doubling dimension d, let $\mathcal{D}$ be a hierarchical decomposition of the metric, and $\varepsilon > 0$. Let also L be a solution to the instance for any of the problems Facility Location, k-Median, or k-Means. A client $v \in C$ is* badly cut w.r.t. $\mathcal{D}$ *if the ball $\beta(v, 3L_v/\varepsilon)$ is cut as some level j greater than $\log(3L_v/\varepsilon) + \tau(\varepsilon, d)$, where L_v is the distance from v to the closest facility of L.*

Similarly, a center $f \in F$ of L is badly cut w.r.t. $\mathcal{D}$ *if $\beta(f, 3\mathrm{OPT}_f)$ is cut at some level j greater than $\log(3\mathrm{OPT}_f) + \tau(\varepsilon, d)$, where OPT_f is the distance from f to the closest facility of* OPT.

In the following, when $\mathcal{D}$ is clear from the context we simply say badly cut. The following lemma bounds the probability of being badly cut.

Lemma 2.5. *Let $(C \cup F, \mathrm{dist})$ be a metric, and $\mathcal{D}$ a random hierarchical decomposition given by Lemma 2.3. Let v be a vertex in $C \cup F$. The probability that v is badly cut is at most $\kappa(\varepsilon, p)$.*

Proof. Consider first a vertex $v \in C$. By Property 1, the probability that a ball $\beta(v, r)$ is cut at level at least j is at most $2^{2d+2} r / 2^j$. Hence the probability that a ball $\beta(v, 3L_v/\varepsilon)$ is cut at a level j greater than $\log(3L_v/\varepsilon) + 2 + 2d + \log(1/\kappa(\varepsilon, p))$ is at most $\kappa(\varepsilon, p)$.

The proof for $v \in F$ is identical. □

2.4 Preprocessing

In the following, we will work with the slightly more general version of the clustering problems where there is some *demand* on each vertex: there is a function $\chi : C \mapsto \{1, \dots, n\}$ and the goal is to minimize $\sum_{c \in C} \chi(c) \cdot \min_{f \in S} \operatorname{dist}(c, f) + \sum_{f \in S} w_f$ for the Facility Location problem, or $\sum_{c \in C} \chi(c) \cdot \min_{f \in S} \operatorname{dist}(c, f)$ and $\sum_{c \in C} \chi(c) \cdot \min_{f \in S} \operatorname{dist}(c, f)^2$ for k-Median and k-Means respectively. This also extends to any $\sum_{c \in C} \chi(c) \cdot \min_{f \in S} \operatorname{dist}(c, f)^p$ with constant p. For simplicity, we will consider in the proof that the client set is actually a multiset, where a client c appears $\chi(c)$ times.

We will preprocess the input instance to transform it into several instances of the more general clustering problem, ensuring that the aspect-ratio Δ of each instance is polynomial. We defer this construction to Appendix A.1.

3 A Near-Linear Time Approximation Scheme for Non-Uniform Facility Location

To demonstrate the utility of the notion of badly cut, we show how to use it to get a near-linear time approximation scheme for Facility Location in metrics of bounded doubling dimension. In this context we refer to centers in the set F of the input as facilities.

We first show a structural lemma that allows to focus on instances that do not contain any badly cut client. Then, we prove that these instances have *portal-respecting* solutions that are nearly optimal, and that can be computed with a dynamic program. We conclude by providing a fast dynamic program, that takes advantage of all the structure provided before.

3.1 An instance with small distortion

Consider a metric space (V, dist) and an instance $\mathcal{I}$ of the Facility Location problem on (V, dist). Here we generalize slightly, as explained in Section 2.4, and restrict the set of candidate center to a subset C of V. Our first step is to show that, given $\mathcal{I}$, a randomized decomposition $\mathcal{D}$ of (V, dist) and any solution L for $\mathcal{I}$ on (V, dist), we can build an instance $\mathcal{I}_{\mathcal{D}}$ on the same metric (but different client and center sets) such that any solution S has a similar cost in $\mathcal{I}$ and in $\mathcal{I}_{\mathcal{D}}$, and more importantly $\mathcal{I}_{\mathcal{D}}$ does not contain any badly cut client with respect to $\mathcal{D}$. The definition of $\mathcal{I}_{\mathcal{D}}$ depends on the randomness of $\mathcal{D}$.

Let $\operatorname{cost}_{\mathcal{I}_0}(S) = \sum_{c \in C} \min_{f \in S}(\operatorname{dist}(c, f))^p$ be the cost incurred by only the distances to the facilities in a solution S to an instance $\mathcal{I}_0$, and let $\varepsilon > 0$. For any instance $\mathcal{I}_{\mathcal{D}}$ on (V, dist), we let

$$\nu_{\mathcal{I}_{\mathcal{D}}} = \max_{\text{solution } S} \left\{ \operatorname{cost}_{\mathcal{I}}(S) - (1 + 2\varepsilon) \operatorname{cost}_{\mathcal{I}_{\mathcal{D}}}(S), (1 - 2\varepsilon) \operatorname{cost}_{\mathcal{I}_{\mathcal{D}}}(S) - \operatorname{cost}_{\mathcal{I}}(S) \right\}.$$

Note that in the particular case of Facility Location, $p = 1$, but we allow it to be more general in order to make the proof adjustable to k-Means. If $B_{\mathcal{D}}$ denotes the set of badly

cut facilities (w.r.t $\mathcal{D}$) of the solution L from which instance $\mathcal{I}_\mathcal{D}$ is constructed, we say that $\mathcal{I}_\mathcal{D}$ has *small distortion w.r.t.* $\mathcal{I}$ if $\sum_{f\in B_\mathcal{D}} w_f \leq \varepsilon \cdot \sum_{f\in L} w_f$, and $\nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon \text{cost}_\mathcal{I}(L)$, and there exists a solution S that contains $B_\mathcal{D}$ with

$$\text{cost}_{\mathcal{I}_\mathcal{D}}(S) \leq (1 + O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT}) + O(\varepsilon)\text{cost}_\mathcal{I}(L). \tag{1}$$

When $\mathcal{I}$ is clear from the context we simply say that $\mathcal{I}_\mathcal{D}$ has small distortion. We will show that the solution $\text{OPT}' = \text{OPT} \cup B_\mathcal{D}$ (where OPT is the optimal solution for the instance $\mathcal{I}$) fulfills the condition of (1).

In the following, we will always work with a particular $\mathcal{I}_\mathcal{D}$ constructed from $\mathcal{I}$ and a precomputed approximate solution L for $\mathcal{I}$ as follows: $\mathcal{I}$ is transformed such that every badly cut client c is moved to $L(c)$, namely, $\chi(L(c))$ is increased by $\chi(c)$ after which we set $\chi(c) = 0$. Recall however that we treat the client set as a multiset, so that $\text{cost}_{\mathcal{I}_0}(S)$ counts the distance from c to the closest facility $\chi(c)$ times.

What we would like to prove is that the optimal solution in $\mathcal{I}$ can be transformed to a solution in $\mathcal{I}_\mathcal{D}$ with a small additional cost, and vice versa. The intuition behind this is the following: a client of the solution L is badly cut with probability $\kappa(\varepsilon, p)$ (from Lemma 2.5), hence every client contributes with $\kappa(\varepsilon, p)L_c$ to transform any solution S for the instance $\mathcal{I}$ to a solution for the instance $\mathcal{I}_\mathcal{D}$, and vice versa.

However, we will need to convert a particular solution in $\mathcal{I}_\mathcal{D}$ (think of it as $\text{OPT}_{\mathcal{I}_\mathcal{D}}$) to a solution in $\mathcal{I}$: this particular solution depends in the randomness of $\mathcal{D}$, and this short argument does not apply because of dependency issues. It is nevertheless possible to prove that $\mathcal{I}_\mathcal{D}$ has a small distortion, as done in the following lemma.

Lemma 3.1. *Given an instance $\mathcal{I}$ of Facility Location, a randomized decomposition $\mathcal{D}$, an ε such that $0 < \varepsilon < 1/4$ and a solution L, let $\mathcal{I}_\mathcal{D}$ be the instance obtained from $\mathcal{I}$ by moving every badly cut client c to $L(c)$ (as described above). The probability that $\mathcal{I}_\mathcal{D}$ has small distortion is at least $1 - \varepsilon$, where the solution fulfilling (1) is $\text{OPT}' = \text{OPT} \cup B_\mathcal{D}$.*

Proof. To show the lemma, we will show that $\mathbb{E}\left[\sum_{f\in B_\mathcal{D}} w_f\right] \leq \varepsilon^2 \sum_{f\in L} w_f/2$ and $\mathbb{E}[\,\nu_{\mathcal{I}_\mathcal{D}}\,] \leq \varepsilon^2\text{cost}(L)/2$. Then, Markov's inequality and a union bound over the probabilities of failure yield $\sum_{f\in B_\mathcal{D}} w_f \leq \varepsilon \cdot \sum_{f\in L} w_f$ and $\nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon\text{cost}_\mathcal{I}(L)$. Since $\text{OPT} \subseteq \text{OPT}'$ and $(1 - 2\varepsilon)\text{cost}_{\mathcal{I}_\mathcal{D}}(\text{OPT}') - \text{cost}_\mathcal{I}(\text{OPT}') \leq \nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon\text{cost}_\mathcal{I}(L)$, the cost incurred by connecting clients to facilities in OPT' is

$$\begin{aligned}
\text{cost}_{\mathcal{I}_\mathcal{D}}(\text{OPT}') &\leq (1 + \frac{2\varepsilon}{1 - 2\varepsilon})(\text{cost}_\mathcal{I}(\text{OPT}') + \varepsilon\text{cost}_\mathcal{I}(L)) && (\text{as } \nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon\text{cost}_\mathcal{I}(L)) \\
&< (1 + 4\varepsilon)(\text{cost}_\mathcal{I}(\text{OPT}') + \varepsilon\text{cost}_\mathcal{I}(L)) && (\text{as } \varepsilon < 1/4) \\
&\leq (1 + 4\varepsilon)(\text{cost}_\mathcal{I}(\text{OPT}) + \varepsilon\text{cost}_\mathcal{I}(L)) && (\text{as OPT} \subseteq \text{OPT}'),
\end{aligned}$$

which shows that $\mathcal{I}_\mathcal{D}$ has small distortion.

Note that $\mathbb{E}\left[\sum_{f\in B_{\mathcal{D}}} w_f\right] = \sum_{f\in L}\Pr[f \text{ badly cut}]\cdot w_f \leq \varepsilon^2\sum_{f\in L} w_f/2$ is immediate from Lemma 2.5. It remains to show that $\mathbb{E}\left[\nu_{\mathcal{I}_{\mathcal{D}}}\right] \leq \varepsilon^2\text{cost}(L)/2$. For the sake of lightening equations, we will denote by $\sum_{\text{bcc. } c}$ the sum over all badly cut clients c.

By definition, we have that for any solution S,

$$\begin{aligned}\text{cost}(S) - \text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) &\leq \sum_{\text{bcc. } c} \text{dist}(c,S)^p - \text{dist}(S,L(c))^p \\ &\leq \sum_{\text{bcc. } c} \big((1+\varepsilon/p)^p\text{dist}(S,L(c))^p \\ &\qquad + (1+p/\varepsilon)^p\text{dist}(c,L(c))^p - \text{dist}(S,L(c))^p\big)\end{aligned}$$

using Lemma 2.2 with parameter $x = \varepsilon/p$.

To bound $(1+\varepsilon/p)^p$, we use that $\forall x \leq 1/p,\ (1+x)^p \leq \exp(xp) \leq 1 + px + e(px)^2/2$. Hence, since $\varepsilon \leq 1/4$, $(1+\varepsilon/p)^p \leq 1 + \varepsilon + e\varepsilon^2/2 \leq 1 + 2\varepsilon$. Plugging this into the right hand side, we get

$$\begin{aligned}\text{cost}(S) - \text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) &\leq \sum_{\text{bcc. } c} \big((1+2\varepsilon)\text{dist}(S,L(c))^p \\ &\qquad + (1+p/\varepsilon)^p\text{dist}(c,L(c))^p - \text{dist}(S,L(c))^p\big) \\ &= \sum_{\text{bcc. } c} 2\varepsilon\cdot\text{dist}(S,L(c))^p + (1+p/\varepsilon)^p\text{dist}(c,L(c))^p.\end{aligned}$$

Subtracting $\sum_{\text{bcc. } c} 2\varepsilon\cdot\text{dist}(S,L(c))^p \leq \sum_{c\in C} 2\varepsilon\cdot\text{dist}(S,L(c))^p$ from the right and left side, respectively, yields

$$\text{cost}(S) - (1+2\varepsilon)\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) \leq \sum_{\text{bcc. } c} (1+p/\varepsilon)^p\text{dist}(c,L(c))^p$$

Similarly, we have that

$$\begin{aligned}\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) - \text{cost}(S) &\leq \sum_{\text{bcc. } c} \text{dist}(S,L(c))^p - \text{dist}(c,S)^p \\ &\leq \sum_{\text{bcc. } c} \big((1+2\varepsilon)\text{dist}(c,S)^p \\ &\qquad + (1+p/\varepsilon)^p\text{dist}(c,L(c))^p - \text{dist}(c,S)^p\big) \\ &\leq \sum_{\text{bcc. } c} 2\varepsilon\cdot\text{dist}(c,S)^p + (1+p/\varepsilon)^p\text{dist}(c,L(c))^p\end{aligned}$$

and we conclude

$$(1-2\varepsilon)\text{cost}_{\mathcal{I}_\mathcal{D}}(S) - \text{cost}(S) \leq \sum_{\text{bcc. } c} (1+p/\varepsilon)^p \text{dist}(c, L(c))^p$$

Therefore, the expected value of $\nu_{\mathcal{I}_\mathcal{D}}$ is

$$E[\nu_{\mathcal{I}_\mathcal{D}}] \leq \sum_{\text{client } c} Pr[c \text{ badly cut}] \cdot (1+p/\varepsilon)^p \text{dist}(c, L(c))^p.$$

Applying Lemma 2.5 and using $\kappa(\varepsilon, p) = \varepsilon^2(\frac{p}{p+\varepsilon})^p$, we conclude $E[\nu_{\mathcal{I}_\mathcal{D}}] \leq \varepsilon^2 \cdot \text{cost}(L)$. The lemma follows for a sufficiently small ε. □

3.2 Portal Respecting Solution

In the following, we fix an instance $\mathcal{I}$, a decomposition $\mathcal{D}$, and a solution L. By Lemma 3.1, $\mathcal{I}_\mathcal{D}$ has small distortion with probability at least $1-\varepsilon$ and so we condition on this event from now on.

We explore the structure that this conditioning can give to solutions. We will show that in the solution $\text{OPT}' = \text{OPT} \cup B_\mathcal{D}$ with small cost, each client c is cut from its serving facility f at a level at most $\log(3L_c/\varepsilon + 4\text{OPT}_c)) + \tau(\varepsilon, d)$. This will allow us to consider *portal-respecting* solution, where every client to facility path goes in and out parts of the decomposition only at designated portals. Indeed, the detour incurred by using a portal-respecting path instead of a direct connection depends on the level where its extremities are cut, as proven in Lemma A.3. Hence, ensuring that this level stays small implies that the detour made is small (in our case, $O(\varepsilon(L_c + \text{OPT}_c))$. Such a solution can be computed by a dynamic program that we will present afterwards.

Recall that L_c and OPT_c are the distances from the *original* position of c to L and OPT, although c may have been moved to $L(c)$ and $B_\mathcal{D}$ is the set of badly cut facilities of L w.r.t $\mathcal{D}$.

Lemma 3.2. *Let $\mathcal{I}$ be an instance of Facility Location with a randomized decomposition $\mathcal{D}$, $\varepsilon < 1/4$ and L be a solution for $\mathcal{I}$, such that $\mathcal{I}_\mathcal{D}$ has small distortion. Let $\text{OPT}' = \text{OPT} \cup B_\mathcal{D}$, and for any client c in $\mathcal{I}_\mathcal{D}$, let $\text{OPT}'(c)$ be the closest facility to c in OPT'. Then c and $\text{OPT}'(c)$ are cut in $\mathcal{D}$ at level at most $\log(3L_c/\varepsilon + 4\text{OPT}_c) + \tau(\varepsilon, d)$.*

Proof. Let c be a client. To find the level at which c and $\text{OPT}'(c)$ are separated, we distinguish between two cases: either c in $\mathcal{I}$ is badly cut w.r.t. $\mathcal{D}$, or not.

If c is badly cut, then it is now located at $L(c)$ in the instance $\mathcal{I}_{\mathcal{D}}$. In that case, either:

1. $L(c)$ is also badly cut, and therefore $L(c) \in B_{\mathcal{D}} \subseteq \mathrm{OPT}'$ and so $\mathrm{OPT}'(c) = L(c)$. Since c and $L(c)$ are collocated, it follows that c and $\mathrm{OPT}'(c)$ are never cut.

2. $L(c)$ is not badly cut: Definition 2.4 implies that $L(c)$ and $\mathrm{OPT}(L(c))$ are cut at a level at most $\log(3\mathrm{OPT}_{L(c)}) + \tau(\varepsilon, d)$. By triangle inequality, $\mathrm{OPT}_{L(c)} \leq L_c + \mathrm{OPT}_c$, and thus c (located at $L(c)$ in $\mathcal{I}_{\mathcal{D}}$) and $\mathrm{OPT}'(c)$ are also cut at level at most $\log\big(3L_c + 3\mathrm{OPT}_c\big) + \tau(\varepsilon, d)$.

We now turn to the case where c is not badly cut. In this case c is not moved to $L(c)$ and the ball $\beta(c, 3L_c/\varepsilon)$ is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$. We make a case distinction according to OPT_c and L_c.

1. If $L_c \leq \varepsilon \mathrm{OPT}_c$, then we have the following. If $L(c)$ is badly cut, $L(c)$ is open in OPT' and therefore $\mathrm{OPT}'_c = L_c$. Moreover, since c is not badly cut the ball $\beta(c, L_c)$ is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$. Therefore c and $\mathrm{OPT}'(c)$ are cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$.

   Now consider the case where $L(c)$ is not badly cut. Both c and $\mathrm{OPT}'(c)$ lie in the ball centered at $L(c)$ and of diameter $2\mathrm{OPT}_{L(c)}$: indeed, we use $L_c \leq \varepsilon \mathrm{OPT}_c$ to derive

$$\begin{aligned} \mathrm{dist}(c, L(c)) &\leq \varepsilon \mathrm{dist}(c, \mathrm{OPT}(c)) \leq \varepsilon \mathrm{dist}(c, \mathrm{OPT}(L(c))) \\ &\leq \varepsilon \mathrm{dist}(c, L(c)) + \varepsilon \mathrm{dist}\big(L(c), \mathrm{OPT}(L(c))\big), \end{aligned}$$

   and therefore $\mathrm{dist}(c, L(c)) \leq \frac{\varepsilon}{1-\varepsilon}\mathrm{OPT}_{L(c)} \leq 2\mathrm{OPT}_{L(c)}$, since $\varepsilon \leq 1/4$. On the other hand,

$$\begin{aligned} \mathrm{dist}(\mathrm{OPT}'(c), L(c)) &\leq \mathrm{dist}(c, \mathrm{OPT}'(c)) + \mathrm{dist}(c, L(c)) \\ &\leq \mathrm{dist}(c, \mathrm{OPT}(c)) + \mathrm{dist}(c, L(c)) \\ &\leq \mathrm{dist}(c, \mathrm{OPT}(L(c))) + \mathrm{dist}(c, L(c)) \\ &\leq 2\mathrm{dist}(c, L(c)) + \mathrm{dist}(L(c), \mathrm{OPT}(L(c))) \\ &\leq \left(1 + \frac{2\varepsilon}{1-\varepsilon}\right) \mathrm{OPT}_{L(c)}, \end{aligned}$$

   which is smaller than $2\mathrm{OPT}_{L(c)}$ for any $\varepsilon \leq 1/4$. Hence we have $c, \mathrm{OPT}'(c) \in \beta(L(c), 2\mathrm{OPT}_{L(c)})$.

   By definition of badly cut, c and $\mathrm{OPT}'(c)$ are therefore cut at level at most $\log(3\mathrm{OPT}_{L(c)}) + \tau(\varepsilon, d)$. Since $\mathrm{OPT}_{L(c)} \leq \mathrm{dist}(L(c), \mathrm{OPT}(c)) \leq \mathrm{dist}(L(c), c) + \mathrm{dist}(c, \mathrm{OPT}(c)) \leq (1+\varepsilon)\mathrm{OPT}_c$ as $L_c \leq \varepsilon \mathrm{OPT}_c$, we have that $\log(3\mathrm{OPT}_{L(c)}) \leq \log(4\mathrm{OPT}_c)$. Hence c and $\mathrm{OPT}'(c)$ are cut at level at most $\log(4\mathrm{OPT}_c) + \tau(\varepsilon, d)$.

2. If $\mathrm{OPT}_c \leq L_c/\varepsilon$, then since c is not badly cut the ball $\beta(c, L_c/\varepsilon)$ in which lies OPT_c is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$.

In all cases, c and $L(c)$ are cut at level at most $\log(3L_c/\varepsilon+4\mathrm{OPT}_c)+\tau(\varepsilon,d)$. This concludes the proof. □

We then aim at proving that there exists a near-optimal "portal-respecting" solution, as we define below. A *path* between two nodes u and v is a sequence of nodes $w_1,\ldots,w_k$, where $u=w_1$ and $v=w_k$, and its length is $\sum \mathrm{dist}(w_j,w_{j+1})$. A solution can be seen as a set of facilities, together with a path for each client that connects it to a facility, and the cost of the solution is given by the sum over all path lengths. We say that a path $w_1,\ldots,w_k$ is *portal-respecting* if for every pair w_j,w_{j+1}, whenever w_j and w_{j+1} lie in different parts $B,B'\in\mathcal{B}_i$ of the decomposition $\mathcal{D}$ on some level i, then these nodes are also portals at this level, i.e., $w_j,w_{j+1}\in\mathcal{P}_B\cup\mathcal{P}_{B'}$. As explained in Lemma A.3, if two vertices u and v are cut at level i, then there exists a portal-respecting path from u to v of length at most $\mathrm{dist}(u,v)+16\rho 2^i$. We define a *portal-respecting* solution to be a solutions such that each path from a client to its closest facility in the solution is portal-respecting.

The dynamic program will compute an optimal portal-respecting solution. Therefore, we need to prove that the optimal portal-respecting solution is close to the optimal solution. We actually show something slightly stronger. Given a solution S, we define $b(S):=\sum_{c,i:\ c \text{ and } S(c) \text{ cut at level } i}\varepsilon 2^i$: one can see $b(S)$ as a *budget*, given by the fact that vertices are not badly cut. Next we show a structural lemma, that bounds the cost of a structured solution and of its budget.

Lemma 3.3 (Structural lemma)**.** *Given an instance $\mathcal{I}$, an ε such that $0<\varepsilon\le 1/4$ and a solution L, it holds with probability $1-\varepsilon$ (over $\mathcal{D}$) that there exists a portal-respecting solution S in $\mathcal{I}_\mathcal{D}$ such that $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S)+b(S)=(1+O(\varepsilon))\mathrm{cost}_\mathcal{I}(\mathrm{OPT})+O(\varepsilon\mathrm{cost}_\mathcal{I}(L))$.*

Proof. From Lemma 3.1, with probability $1-\varepsilon$ it holds that the instance $\mathcal{I}_\mathcal{D}$ has small distortion, and $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(\mathrm{OPT}')\le(1+4\varepsilon)(\mathrm{cost}_\mathcal{I}(\mathrm{OPT})+\varepsilon\mathrm{cost}_\mathcal{I}(L))$. We now bound the cost of making OPT' portal respecting by applying Lemma 3.2. Since each client c of $\mathcal{I}_\mathcal{D}$ is cut from $\mathrm{OPT}'(c)$ at level at most $\log(3L_c/\varepsilon+4\mathrm{OPT}_c)+\tau(\varepsilon,d)$, we have that the detour for making the assignment of c to $\mathrm{OPT}'(c)$ portal-respecting is at most $O(\rho 2^{\tau(\varepsilon,d)}(L_c/\varepsilon+\mathrm{OPT}_c))$. Choosing $\rho=\varepsilon^2 2^{-\tau(\varepsilon,d)}$ ensures that the detour is at most $O(\varepsilon(L_c+\mathrm{OPT}_c))$. Summing over all clients c gives a total detour of $O(\varepsilon)(\mathrm{cost}_\mathcal{I}(L)+\mathrm{cost}_\mathcal{I}(\mathrm{OPT}))$. The resulting portal respecting tour is the solution S we are looking for. The above calculation also bounds $b(S)=b(\mathrm{OPT}')\le O(\varepsilon)(\mathrm{cost}_\mathcal{I}(L)+\mathrm{cost}_\mathcal{I}(\mathrm{OPT}))$, and so $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S)+b(S)=(1+O(\varepsilon))\mathrm{cost}_\mathcal{I}(\mathrm{OPT})+O(\varepsilon\mathrm{cost}_\mathcal{I}(L))$. □

3.3 The Algorithm

Using Lemma A.1 and A.2, we can assume that the aspect-ratio of the instance is $O(n^5/\varepsilon)$. Our algorithm starts by computing a constant-factor approximation L, using Meyerson's algorithm [35]. It then computes a hierarchical decomposition $\mathcal{D}$, as explained in the Section 2.2, with parameter $\rho=\varepsilon 2^{-\tau(\varepsilon,d)}$.

Given L and the decomposition $\mathcal{D}$, our algorithm finds all the badly cut clients as follows. For each client c, to determine whether c is badly cut or not, the algorithm checks whether the decomposition cuts $\beta(c, 3L_c/\varepsilon)$ at a level higher than $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$, making c badly cut. This can be done efficiently, since c is in exactly one part at each level, by verifying whether c is at distance smaller than $3L_c/\varepsilon$ to such a part of too high level. Thus, the algorithm finds all the badly cut clients in near-linear time.

The next step of the algorithm is to compute instance $\mathcal{I}_{\mathcal{D}}$ by moving every badly cut client c to its facility in L. This can be done in linear time.

A first attempt at a dynamic program. We now turn to the description of the dynamic program (DP) for obtaining the best portal-respecting solution of $\mathcal{I}_{\mathcal{D}}$. This is the standard dynamic program for Facility Location and we only describe it for the sake of completeness. The reader familiar with this can therefore skip to the analysis.

There is a table entry for each part of the decomposition, and two vectors of length $|\mathcal{P}_B|$, where $\mathcal{P}_B$ is the set of portals in the part B. We call such a triplet a configuration. Each configuration given by a part B and vectors $\langle \ell_1, \ldots, \ell_{|\mathcal{P}_B|} \rangle$ and $\langle s_1, \ldots, s_{|\mathcal{P}_B|} \rangle$ (called the *portal parameters*), encodes a possible interface between part B and a solution for which the ith portal has distance ℓ_i to the closest facility inside of B, and distance s_i to its closest facility outside of B. The value stored for such a configuration in a table entry is the minimal cost for a solution with facilities respecting the constraints induced by the vectors on the distances between the solution and the portals inside the part (as described below).

To fill the table, we use a dynamic program following the lines of Arora et al. [4] or Kolliopoulos and Rao [25]. If a part has no descendant (meaning the part contains a single point), computing the solution given the configuration is straightforward: either a center is opened on this point or not, and it is easy to check the consistency with the configuration, where only the distances to portals inside the part need to be verified. At a higher level of the decomposition, a solution is simply obtained by going over all the sets of parameter values for all the children parts. It is immediate to see whether sets of parameter values for the children can lead to a consistent solution:

- for each portal p_1 of the parent part, there must be one portal p_2 of a child part such that the distance from p_1 to a center inside the part prescribed by the configuration corresponds to $\text{dist}(p_1, p_2)$ plus the distance from p_2 to a center inside the child part;
- for each portal p_2 of a child part, there must exist either:
  - a portal p_1 of the parent part such that the distance from p_2 to a center outside its part prescribed by the configuration is $\text{dist}(p_1, p_2)$ plus the distance from p_1 to a center outside of the part,
  - or a portal p_1 of another child part such that this distance is $\text{dist}(p_1, p_2)$ plus the distance from p_1 to a center inside the child part.

The runtime of this algorithm depends on the number of possible distances determining the number of possible portal parameters. Even if the aspect ratio is polynomial, there can be a large number of possible distances, so that the number of configurations might be exponential. Using the budget given by Lemma 3.3, one can approximate the distances and obtain an efficient algorithm, as we show next.

A faster dynamic program. We now describe a faster dynamic program. Consider a level where the diameter of the parts is say D. Each configuration is again given by a part B and portal parameters $\langle \ell_1, \ldots, \ell_{|\mathcal{P}_B|} \rangle$ and $\langle s_1, \ldots, s_{|\mathcal{P}_B|} \rangle$, but with the restriction that ℓ_i and s_i are multiples of εD in the range $[0, D/\varepsilon + D]$. A boolean flag is additionally attached to the configuration (whose meaning will be explained shortly).

We sketch here the intuition behind this restriction. Since the diameter of the part is D we can afford a detour of εD, that can be charged to the budget $b(S)$. Hence, distances can be rounded to the closest multiple of εD.

Now, suppose that the closest facility outside the part is at distance greater than D/ε, and that there is no facility inside the part. Then, since the diameter is D, up to losing an additive $D \leq \varepsilon \mathrm{OPT}$ in the cost of the solution computed, we may assume that all the points of the part are assigned to the same facility. So the algorithm is not required to have the precise distance to the closest center outside the part, and it uses the flag to reflect that it is in this regime. We can then treat this whole part as a single client (weighted by the number of clients inside the part) to be considered at higher levels. Assuming that the closest facility is at distance less than D/ε, we have that for any portal of the part the closest facility is at distance at most $D/\varepsilon + D$ (since D is the diameter of the part).

On the other hand, if there is some facility inside the part and the closest facility outside the part is at distance at least D/ε, then each client of the part should be served by a facility inside the part in any optimal assignment. Thus it is not necessary that the algorithm iterates over configurations where the distances outside the part are more than D/ε: it is enough to do it once and use the answer for all other queries.

Analysis – Proof of Theorem 1.3. The following lemmas show that the solution computed by this algorithm is a near-optimal one, and that the complexity is near-linear: this proves Theorem 1.3. We first bound the connection cost in $\mathcal{I}_{\mathcal{D}}$.

Lemma 3.4. *Let S be as in Lemma 3.3. The algorithm computes a solution S^* with cost at most $\mathrm{cost}_{\mathcal{I}_{\mathcal{D}}}(S^*) \leq (1 + O(\varepsilon))\mathrm{cost}_{\mathcal{I}_{\mathcal{D}}}(S) + b(S)$.*

Proof. We show that the solution S can be adapted to a configuration of the DP with extra cost $b(S)$. For this, let c be a client served by a facility $S(c)$, and let $w_1, \ldots, w_k$ be the portal-respecting path from c to $S(c)$ with $w_1 = c$ and $w_k = S(c)$. The cost contribution of c to S is therefore $\sum_{i=1}^{k-1} \mathrm{dist}(w_i, w_{i+1})$. For each w_i, let also l_i be the level at which w_i is a portal.

The distance between c and $S(c)$ is approximated at several places of the DP. Consider any node w_i on the path from c to $S(c)$:

- When $\text{dist}(w_i, S(c)) \leq 2^{l_i}/\varepsilon + 2^{l_i}$, the distance between w_i and $S(c)$ is rounded to the closest multiple of $\varepsilon 2^{l_i}$, incurring a cost increase of $\varepsilon 2^{l_i}$.
- When $\text{dist}(w_i, S(c)) \geq 2^{l_i}/\varepsilon + 2^{l_i}$, the whole part is contracted and served by a single facility at distance at least $2^{l_i}/\varepsilon$. The cost difference for client c is therefore 2^{l_i}. Since the diameters of the parts are geometrically increasing, the total cost difference for all contractions of regions containing c is bounded by 2^{l_j+1}, where l_j is the highest level where $d(w_j, S(c)) \geq 2^{j_i}/\varepsilon + 2^{l_j}$. This inequality implies that $2^{l_j+1} \leq 2\varepsilon d(w_j, S(c))$, which is smaller than $2\varepsilon \sum d(w_i, w_{i+1})$, the cost of c in the portal-respecting solution S.

Hence, summing over all clients, the additional cost incurred by the DP is at most $b(S) + 4\varepsilon \text{cost}_{\mathcal{I}_\mathcal{D}}(S)$. Since it computes a solution with minimal cost, it holds that $\text{cost}_{\mathcal{I}_\mathcal{D}}(S^*) \leq (1 + 4\varepsilon)\text{cost}_{\mathcal{I}_\mathcal{D}}(S) + b(S)$. □

Next we bound the connection cost in $\mathcal{I}$. If $\mathcal{I}_\mathcal{D}$ has small distortion, the facility cost increase due to badly cut clients is bounded by $\sum_{f \in B_\mathcal{D}} w_f \leq \varepsilon \sum_{f \in L} w_f$, since we have $\text{OPT}' = \text{OPT} \cup B_\mathcal{D}$. Thus due to the following corollary the total solution cost of S^* is bounded as required.

Corollary 3.5. *Let S^* be the solution computed by the algorithm. With probability $1 - \varepsilon$, it holds that $\text{cost}_\mathcal{I}(S^*) = (1 + O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT})$.*

Proof. Lemma 3.3 ensures that, with probability $1 - \varepsilon$, the cost of S in $\mathcal{I}_\mathcal{D}$ and $b(S)$ is at most $(1 + O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT}) + O(\varepsilon \text{cost}_\mathcal{I}(L))$. Since L is a constant-factor approximation of OPT in $\mathcal{I}$, this cost turns out to be $(1 + O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT})$. Using that $\mathcal{I}_\mathcal{D}$ has small distortion, and combining this with Lemma 3.4 concludes the proof:

$$\begin{aligned}
\text{cost}_\mathcal{I}(S^*) &\leq (1 + 2\varepsilon)\text{cost}_{\mathcal{I}_\mathcal{D}}(S^*) + \varepsilon \text{cost}_\mathcal{I}(L) \\
&\leq (1 + O(\varepsilon))(\text{cost}_{\mathcal{I}_\mathcal{D}}(S) + b(S)) + \varepsilon \text{cost}_\mathcal{I}(L) \\
&\leq (1 + O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT})
\end{aligned}$$

□

Lemma 3.6. *This algorithm runs in $2^{(1/\varepsilon)^{O(d^2)}} n + 2^{O(d)} n \log n$ time.*

Proof. The preprocessing step (computing L, the hierarchical decomposition $\mathcal{D}$, and the instance $\mathcal{I}_\mathcal{D}$) has a running time $O(n \log n)$, as all the steps can be done with this complexity: a fast implementation of Meyerson's algorithm [35] tailored to graphs can compute L in time $O(m \log n)$. Using it on the spanner with constant stretch computed with [22] gives a $O(1)$-approximation in time $O(n \log n)$. As explained earlier, the hierarchical decomposition $\mathcal{D}$ and the instance $\mathcal{I}_\mathcal{D}$ can also be computed with this complexity. The decomposition

can moreover be transformed in order to remove part that do not contain any point, as well as degree 2 nodes. This ensures to have $O(n)$ part in total, since there are n leaves and a degree at least 3.

The DP has a linear time complexity: in a part of diameter D, the portal set is a $(\varepsilon 2^{-\tau(\varepsilon,d)}D)$-net, and hence has size $2^{d\lceil \log(2^{\tau(\varepsilon,d)}/\varepsilon)\rceil}$ by Lemma 2.1. Since $\tau(\varepsilon,d) = 2d + 2 + \log\frac{(p+\varepsilon)^p}{\varepsilon^2 p^p}$, this number can be simplified to $2^{O(d^2+d\log(1/\varepsilon))}$. Since each portal stores a distance that can take only $1/\varepsilon^2$ values, there are at most $T = (1/\varepsilon^2)^{2^{O(d^2+d\log(1/\varepsilon))}} = 2^{2^{O(d^2+d\log(1/\varepsilon))}}$ possible table entries for a given part.

To fill the table, notice that a part has at most $2^{O(d)}$ children, due to the properties of the hierarchical decomposition. For any given part, going over all the sets of parameter values for all the children parts therefore takes time $T^{2^{O(d)}} = 2^{2^{O(d^2+d\log(1/\varepsilon))}}$. This dominates the complexity of computing all table entry for one part of the decomposition.

Since the hierarchical decomposition is a tree with n leaves (one per vertex) and without degree-one internal nodes (those can be compressed), there are at most n parts in the decomposition: the complexity of the dynamic program is therefore $n \cdot 2^{2^{O(d^2+d\log(1/\varepsilon))}}$,

The total complexity of the algorithm is thus

$$n \cdot 2^{2^{O(d^2+d\log(1/\varepsilon))}} + 2^{O(d)} n \log n \qquad \square$$

4 The k-Median and k-Means Problems

We aim at using the same approach as for Facility Location. We focus the presentation on k-Median, and only later show how to adapt the proof for k-Means.

We will work with the more general version of k-Median as defined in Section 2.4, where the instance consists of a set of clients C, a set of candidate centers F, an integer k, and a function $\chi : C \mapsto \{1,\dots,n\}$ and the goal is to minimize $\sum_{c\in C} \chi(c) \cdot \min_{f\in S} \operatorname{dist}(c,f)$. We will consider in the proof that C is actually a multiset, instead of carrying along the multiplicity χ.

The road-map is as for Facility Location: we show in Lemma 4.2 that an instance $\mathcal{I}_\mathcal{D}$ has a *small distortion* with good probability, and then in Lemma 4.5 that if an instance has small distortion then there exists a near-optimal portal-respecting solution. We finally present a dynamic program that computes such a solution.

A key ingredient of the proof for Facility Location was our ability to add all badly-cut facilities to the solution OPT$'$. This is not directly possible in the case of k-Median and k-Means, as the number of facilities is fixed. Hence, the first step of our proof is to show that one can make some room in OPT, by removing a few centers without increasing the cost by too much.

4.1 Towards a Structured Near-Optimal Solution

Let OPT be an optimal solution to $\mathcal{I}$ and L and approximate solution. We consider the mapping of the facilities of OPT to L defined as follows: for any $f \in \text{OPT}$, let $L(f)$ denote the facility of L that is the closest to f. Recall that for a client c, $L(c)$ is the facility serving c in L.

For any facility ℓ of L, define $\psi(\ell)$ to be the set of facilities of OPT that are mapped to ℓ, namely, $\psi(\ell) = \{f \in \text{OPT} \mid L(f) = \ell\}$. Define L^1 to be the set of facilities ℓ of L for which there exists a unique $f \in \text{OPT}$ such that $L(f) = \ell$, namely $L^1 = \{\ell \in L \mid |\psi(\ell)| = 1\}$. Let $L^0 = \{\ell \in L \mid |\psi(\ell)| = 0\}$, and $L^{\geq 2} = L - (L^1 \cup L^0)$. Similarly, define $\text{OPT}^1 = \{f \in \text{OPT} \mid L(f) \in L^1\}$ and $\text{OPT}^{\geq 2} = \{f \in \text{OPT} \mid L(f) \in L^{\geq 2}\}$. Note that $|\text{OPT}^{\geq 2}| = |L^0| + |L^{\geq 2}|$, since $|\text{OPT}^1| = |L^1|$ and, w.l.o.g., $|\text{OPT}| = |L| = k$.

The construction of a structured near-optimal solution is made in 3 steps. The first one defines a solution OPT' as follows. Start with $\text{OPT}' = \text{OPT}$.

- **Step 1.** For each facility $\ell \in L^{\geq 2}$, fix one in $\text{OPT}^{\geq 2}$ that is closest to ℓ, breaking ties arbitrarily, and call it f_ℓ. Let $\mathcal{H} \subseteq \text{OPT}^{\geq 2}$ be the set of facilities of $\text{OPT}^{\geq 2}$ that are not the closest to their corresponding facility in $L^{\geq 2}$, i.e., $f \in \mathcal{H}$ if and only if $f \in \psi(\ell)$ and $f \neq f_\ell$ for some $\ell \in L^{\geq 2}$. Among the facilities of $\mathcal{H}$, remove from OPT' the subset of size $\lfloor \varepsilon \cdot |\text{OPT}^{\geq 2}|/2 \rfloor$ that yields the smallest cost increase. Note that this subset is well-defined if $\varepsilon \leq 1$.

This step makes room to add the badly cut facilities without violating the constraint on the maximum number of centers, while at the same time ensures near-optimal cost, as the following lemma shows.

Lemma 4.1. *After Step 1,* OPT' *has cost* $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(L)$

Proof. We claim that for a client c served by $f \in \mathcal{H}$ in the optimum solution OPT, i.e., $f = \text{OPT}(c)$, the detour entailed by the deletion of f is $O(\text{OPT}_c + L_c)$. Indeed, let f' be the facility of OPT that is closest to $L(f)$, and recall that $L(c)$ is the facility that serves c in the solution L. Since $f' \notin \mathcal{H}$, the cost to serve c after the removal of f is at most $\text{dist}(c, f')$, which can be bounded by $\text{dist}(c, f') \leq \text{dist}(c, f) + \text{dist}(f, L(f)) + \text{dist}(L(f), f')$. But by definition of f', $\text{dist}(f', L(f)) \leq \text{dist}(L(f), f)$, and by definition of the function L we have $\text{dist}(L(f), f) \leq \text{dist}(L(c), f)$, so that $\text{dist}(c, f') \leq \text{dist}(c, f) + 2\text{dist}(f, L(c))$. Using the triangle inequality finally gives $\text{dist}(c, f') \leq 3\text{dist}(c, f) + 2\text{dist}(c, L(c))$ which is $O(\text{OPT}_c + L_c)$. For a facility f of OPT, we denote by $C(f)$ the set of clients served by f, i.e. $C(f) = \{c \in C \mid \text{OPT}(c) = f\}$. The total cost incurred by the removal of f is then $\sum_{c \in C(f)} O(\text{OPT}_c + L_c)$, and the cost of removing all of $\mathcal{H}$ is $O(\text{cost}(\text{OPT}) + \text{cost}(L))$.

Recall that in Step 1 we remove the set $\widehat{\mathcal{H}}$ of size $\lfloor \varepsilon |\text{OPT}^{\geq 2}|/2 \rfloor$ from $\mathcal{H}$, such that $\widehat{\mathcal{H}}$ minimizes the cost increase. We use an averaging argument to bound the cost increase: the sum among all facilities $f \in \mathcal{H}$ of the cost of removing the facility f is less than

$O(\text{cost}(\text{OPT}) + \text{cost}(L))$, and $|\mathcal{H}| = O(1/\varepsilon) \cdot \lfloor \varepsilon |\text{OPT}^{\geq 2}| \rfloor$. Therefore removing $\widehat{\mathcal{H}}$ increases the cost by at most $O(\varepsilon)(\text{cost}(\text{OPT}) + \text{cost}(L))$, so that Step 1 is not too expensive. □

We can therefore use this solution OPT' as a proxy for the optimal solution, and henceforth we will denote this solution by OPT. In particular, the badly cut facilities are defined for this solution and not the original OPT.

4.2 An instance with small distortion

As in Section 3, the algorithm computes a randomized hierarchical decomposition $\mathcal{D}$, and transforms the instance of the problem: every badly cut client c is moved to $L(c)$, namely, there is no more client at c and we add an extra client at $L(c)$. Again, we let $\mathcal{I}_{\mathcal{D}}$ denote the resulting instance and note that $\mathcal{I}_{\mathcal{D}}$ is a random variable that depends on the randomness of $\mathcal{D}$.

Moreover, similar as for Facility Location, we let $B_{\mathcal{D}}$ be the set of centers of L that are badly cut from OPT, i.e., $f \in B_{\mathcal{D}}$ if the ball $\beta(f, 3\text{OPT}_f)$ is cut at some level greater than $\log(3\text{OPT}_f) + \tau(\varepsilon, d)$. We call $\text{cost}_{\mathcal{I}}(S)$ the cost of a solution S in the original instance $\mathcal{I}$, and $\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S)$ its cost in $\mathcal{I}_{\mathcal{D}}$. We let

$$\nu_{\mathcal{I}_{\mathcal{D}}} = \max_{\text{solution } S} \big\{ \text{cost}_{\mathcal{I}}(S) - (1+2\varepsilon)\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S), (1-2\varepsilon)\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) - \text{cost}_{\mathcal{I}}(S) \big\}.$$

We say that an instance $\mathcal{I}_{\mathcal{D}}$ has *small distortion* if $\nu_{\mathcal{I}_{\mathcal{D}}} \leq \varepsilon \text{cost}_{\mathcal{I}}(L)$, and there exists a solution S that contains $B_{\mathcal{D}}$ with $\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) \leq (1+O(\varepsilon))\text{cost}_{\mathcal{I}}(\text{OPT}) + O(\varepsilon)\text{cost}_{\mathcal{I}}(L)$. That is, the condition is the same as for Facility Location, except that we do not need a bound on the opening costs. In contrast to the Facility Location problem, here we need to be more careful when identifying the solution fulfilling the latter inequality.

For this, we go on with the next two steps of our construction, defining a solution S^*. Recall that we defined $f_\ell \in \text{OPT}^{\geq 2}$ to be the closest facility to $\ell \in L^{\geq 2}$, breaking ties arbitrarily. For any $\ell \in L^1$, we also denote by $f_\ell \in \text{OPT}^1$ the unique facility closest to ℓ. We start with $S^* = \text{OPT}$ obtained from Step 1. Note that for every $\ell \in L^1 \cup L^{\geq 2}$ the closest facility $f_\ell \in \text{OPT}$ is still present after Step 1, since only some of the other facilities in $\mathcal{H}$ were removed.

- **Step 2.** For each badly-cut facility $\ell \in B_{\mathcal{D}} - L^0$ (i.e., $\psi(\ell) \neq \emptyset$), replace f_ℓ by ℓ in S^*.
- **Step 3.** Add all badly cut facilities of L^0 to S^*.

We show next that S^* satisfies the conditions for $\mathcal{I}_\mathcal{D}$ to have small distortion with good probability.

Lemma 4.2. *The probability that $\mathcal{I}_\mathcal{D}$ has small distortion is at least $1-\varepsilon$, if $\varepsilon \leq 1/4$.*

Proof. The proof that $\nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon \text{cost}_\mathcal{I}(L)$ with probability at least $1-\varepsilon/2$ is identical to the one in Lemma 3.1. We thus turn to bound the probability that solution S^* satisfies the cardinality and cost requirements. Our goal is to show that this happens with probability at least $1-\varepsilon/2$. Then, taking a union bound over the probabilities of failure yields the proposition.

By Steps 2 and 3, we have that S^* contains $B_\mathcal{D}$. We split the proof of the remaining properties into the following claims.

Claim 4.3. *With probability at least $1-\varepsilon/4$, the set S^* is an admissible solution, i.e., $|S^*| \leq k$.*

Proof. We let b be the number of facilities of L^0 that are badly cut. By Lemma 2.5, we have that $\mathbb{E}[b] \leq \varepsilon^2 |L^0|/4$ as $p=1$. By Markov's inequality, the probability that $b > \lfloor \varepsilon |L^0|/2 \rfloor$ is at most $\varepsilon/2$. Now, condition on the event that $b \leq \lfloor \varepsilon |L^0|/2 \rfloor$. Since $|L^0| + |L^{\geq 2}| = |\text{OPT}^2|$, we have that $b \leq \lfloor \varepsilon |\text{OPT}^{\geq 2}|/2 \rfloor$. Moreover, the three steps converting OPT into S^* ensure that $|S^*| \leq k + b - \lfloor \varepsilon |\text{OPT}^{\geq 2}|/2 \rfloor$, as Step 1 removes $\lfloor \varepsilon |\text{OPT}^{\geq 2}|/2 \rfloor$ facilities, while Step 2 only swaps facilities so their number does not change, and Step 3 adds b facilities. Combining the two inequalities gives $|S^*| \leq k$. □

Claim 4.4. *If $\varepsilon < 1/4$, then with probability at least $1-\varepsilon/4$, $\text{cost}_{\mathcal{I}_\mathcal{D}}(S^*) \leq (1+O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT}) + O(\varepsilon \cdot \text{cost}_\mathcal{I}(L))$*

Proof. We showed in Lemma 4.1 that the cost increase in $\mathcal{I}$ due to Step 1 is at most $O(\varepsilon)(\text{cost}_\mathcal{I}(\text{OPT}) + \text{cost}_\mathcal{I}(L))$. We will prove below that this implies that also Step 2 leads to a cost increase of $O(\varepsilon)(\text{cost}_\mathcal{I}(\text{OPT}) + \text{cost}_\mathcal{I}(L))$ in $\mathcal{I}$ with good probability. Step 3 can only decrease the cost. Hence we have $\text{cost}_\mathcal{I}(S^*) \leq (1+O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT}) + O(\varepsilon \cdot \text{cost}_\mathcal{I}(L))$. To bound the cost of S^* in $\mathcal{I}_\mathcal{D}$, we use that $(1-2\varepsilon)\text{cost}_{\mathcal{I}_\mathcal{D}}(S^*) - \text{cost}_\mathcal{I}(S^*) \leq \nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon \text{cost}_\mathcal{I}(L)$. The cost incurred by connecting clients to facilities in S^* is

$$\begin{aligned}
\text{cost}_{\mathcal{I}_\mathcal{D}}(S^*) &\leq (1 + \frac{2\varepsilon}{1-2\varepsilon})(\text{cost}_\mathcal{I}(S^*) + \varepsilon \cdot \text{cost}_\mathcal{I}(L)) && (\text{as } \nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon \text{cost}_\mathcal{I}(L)) \\
&\leq (1+4\varepsilon)(\text{cost}_\mathcal{I}(S^*) + \varepsilon \cdot \text{cost}_\mathcal{I}(L)) && (\text{as } \varepsilon \leq 1/4) \\
&\leq (1+O(\varepsilon))\text{cost}_\mathcal{I}(\text{OPT}) + O(\varepsilon \cdot \text{cost}_\mathcal{I}(L)) && (\text{by above bound on } \text{cost}_\mathcal{I}(S^*)).
\end{aligned}$$

To bound the cost increase of Step 2, we first show that starting with OPT and replacing every $f_\ell \in \text{OPT}$ by $\ell \in L^1 \cup L^{\geq 2}$ results in a solution S' of cost $O(\text{cost}_\mathcal{I}(\text{OPT}) + \text{cost}_\mathcal{I}(L))$. For that, let c be a client that in OPT is served by a facility f_ℓ that is closest to some $\ell \in L^1 \cup L^{\geq 2}$. Recall that every facility of OPT that is closest to some facility of $L^1 \cup L^{\geq 2}$

is in OPT, as only some of those from $\mathcal{H}$ are removed in Step 1. Hence if c is served by some $\ell' \in L^1 \cup L^{\geq 2}$ in the solution L, then this facility ℓ' is in S' since it will replace the closest facility $f_{\ell'}$. Thus the cost of serving c in S' is $\text{dist}(c, \ell') = \text{dist}(c, L)$. On the other hand, if c is served by a facility ℓ_0 of L^0 in L, then it is possible to serve it by the facility ℓ that replaces f_ℓ. The serving cost then is $\text{dist}(c, \ell) \leq \text{dist}(c, f_\ell) + \text{dist}(f_\ell, \ell) \leq \text{dist}(c, f_\ell) + \text{dist}(f_\ell, \ell_0)$, using that f_ℓ is the closest facility to ℓ in the last inequality. Using again the triangle inequality, this cost is at most $2\text{dist}(c, f_\ell) + \text{dist}(c, \ell_0)$. Moreover, any client served by a facility of $\mathcal{H}$ in OPT, i.e., which is not the closest to a facility of L, can in S' be served by the same facility as in OPT, with cost $\text{dist}(c, \text{OPT})$. Hence the cost of the obtained solution is at most $2\text{cost}_{\mathcal{I}}(\text{OPT}) + \text{cost}_{\mathcal{I}}(L) \leq O(\text{cost}_{\mathcal{I}}(\text{OPT}) + \text{cost}_{\mathcal{I}}(L))$ by Lemma 4.1 and assuming, say, $\varepsilon \leq 1$.

The probability of replacing f_ℓ by $\ell \in L^1 \cup L^{\geq 2}$ in Step 2 is the probability that ℓ is badly cut. This is $\kappa(\varepsilon, p)$ by Lemma 2.5 (note that this probability is the same whether $B_{\mathcal{D}}$ is defined for OPT or OPT). Finally, with linearity of expectation, the expected cost to add the badly cut facilities $\ell \in L^1 \cup L^{\geq 2}$ instead of their closest facility f_ℓ of OPT in Step 2 is $O(\kappa(\varepsilon, p)(\text{cost}_{\mathcal{I}}(\text{OPT}) + \text{cost}_{\mathcal{I}}(L)))$. Markov's inequality thus implies that the cost of this step is at most $O(\varepsilon)(\text{cost}_{\mathcal{I}}(\text{OPT}) + \text{cost}_{\mathcal{I}}(L))$ with probability $1 - \frac{O(\kappa(\varepsilon,p))}{\varepsilon} \geq 1 - \varepsilon/4$, since $\kappa(\varepsilon, p) \leq \varepsilon^2/2$ in the case of k-Median where $p = 1$, and $\varepsilon < 1/4$. □

Lemma 4.2 follows from taking a union bound over the probabilities of failure of Claim 4.3 and 4.4. □

4.3 Portal respecting solution

We have to prove the same structural lemma as for Facility Location, to say that there exists a portal-respecting solution in $\mathcal{I}_{\mathcal{D}}$ with cost close to $\text{cost}(S^*)$ where S^* is the solution obtained from the three steps above. Recall that for any solution S and client c, S_c is the distances from the *original* position of c to S in $\mathcal{I}$, but c may have been moved to $L(c)$ in $\mathcal{I}_{\mathcal{D}}$. Recall also that OPT is defined after removing some centers in Step 1.

Lemma 4.5. *Let $\mathcal{I}$ be an instance of k-Median with a randomized decomposition $\mathcal{D}$, and L be a solution for $\mathcal{I}$, such that $\mathcal{I}_{\mathcal{D}}$ has small distortion. Let S^* be the solution obtained by applying Steps 1, 2 and 3. Then, for any client c in $\mathcal{I}_{\mathcal{D}}$, c and $S^*(c)$ are cut at level at most $\log(4\text{OPT}_c + 3L_c/\varepsilon) + \tau(\varepsilon, d)$ in $\mathcal{D}$, whenever $\varepsilon \leq 1/5$, where $S^*(c)$ is the closest facility to c in S^*.*

Proof. The proof of this lemma is very similar to the one of Lemma 3.2. However, since some facilities of OPT were removed in Step 2 to obtain S^*, we need to adapt the proof carefully. In particular, we will use the following inequality. Let c be a client. If $\text{OPT}(c)$ was moved in Step 2, it was replaced by facility ℓ for which $\text{OPT}(c) = f_\ell$ and $\text{dist}(\text{OPT}(c), \ell) \leq \text{dist}(\text{OPT}(c), L(c))$, since f_ℓ is the closest facility to ℓ. Using the triangle inequality we obtain $\text{dist}(\text{OPT}(c), L(c)) \leq \text{dist}(c, \text{OPT}(c)) + \text{dist}(c, L(c))$. On the other hand, as $\ell \in S^*$

we get $\text{dist}(c, S^*(c)) \leq \text{dist}(c, \ell) \leq \text{dist}(c, \text{OPT}(c)) + \text{dist}(\text{OPT}(c), \ell)$, again applying the triangle inequality. Putting these inequalities together we obtain

$$\text{dist}(c, S^*(c)) \leq 2\text{dist}(c, \text{OPT}(c)) + \text{dist}(c, L(c)). \tag{2}$$

Furthermore, if $\text{OPT}(c)$ is not moved in Step 2 we have $\text{OPT}(c) \in S^*$, and so Inequality (2) holds trivially as $\text{dist}(c, S^*(c)) \leq \text{dist}(c, \text{OPT}(c))$.

To find the level at which c and $S^*(c)$ are cut, we distinguish between two cases: either c in $\mathcal{I}$ is badly cut w.r.t. $\mathcal{D}$, or not. If c is badly cut, then it is now located at $L(c)$ in the instance $\mathcal{I}_{\mathcal{D}}$. In that case, either:

1. $L(c)$ is also badly cut, i.e., $L(c) \in B_{\mathcal{D}} \subseteq S^*$, and so $S^*(c) = L(c)$. It follows that c and $S^*(c)$ are collocated, thus they are never cut.

2. $L(c)$ is not badly cut. Then, since c is now located at $L(c)$, $\text{dist}(c, S^*(c)) = \text{dist}(L(c), S^*(L(c))$. $\text{OPT}(L(c))$ is not necessarily in S^*: in that case, it was replaced by a facility f that is closer to $\text{OPT}(L(c))$ than $L(c)$, and so $d(L(c), f) \leq 2d(L(c), \text{OPT}(L(c))$. Hence, either if $\text{OPT}(L(c))$ is in S^* or not, it holds that $d(L(c), S^*) \leq 2\text{OPT}_{L(c)}$.

   Since $L(c)$ is not badly cut, the ball $\beta(L(c), 3\text{OPT}_{L(c)})$ is cut at level at most $\log(3\text{OPT}_{L(c)}) + \tau(\varepsilon, d)$. By triangle inequality, $\text{OPT}_{L(c)} = \text{dist}(L(c), \text{OPT}(L(c))) \leq L_c + \text{OPT}_c$, and thus c and $S^*(c)$ are also separated at level at most $\log\big(3L_c + 3\text{OPT}_c\big) + \tau(\varepsilon, d)$.

In the other case where c is not badly cut, the ball $\beta(c, 3L_c/\varepsilon)$ is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$. We make a case distinction according to L_c and OPT_c.

1. If $L_c \leq \varepsilon\text{OPT}_c$, then we have the following. If $L(c)$ is badly cut, $L(c) \in B_{\mathcal{D}} \subseteq S^*$ and therefore $S^*_c \leq L_c$. Moreover, since c is not badly cut the ball $\beta(c, L_c)$ is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$. Therefore c and $S^*(c)$ are cut at a level below $\log(4\text{OPT}_c + 3L_c/\varepsilon) + \tau(\varepsilon, d)$.

   In the case where $L(c)$ is not badly cut, both c and $S^*(c)$ lie in the ball centered at $L(c)$ and of diameter $3\text{OPT}_{L(c)}$, which can be seen as follows. We use $L_c \leq \varepsilon\text{OPT}_c$ to derive

$$\begin{aligned}\text{dist}(c, L(c)) &\leq \varepsilon\text{dist}(c, \text{OPT}(c)) \leq \varepsilon\text{dist}(c, \text{OPT}(L(c))) \\ &\leq \varepsilon\text{dist}(c, L(c)) + \varepsilon\text{dist}\big(L(c), \text{OPT}(L(c))\big)\end{aligned}$$

   And therefore, since $\varepsilon \leq 1/4$, $\text{dist}(c, L(c)) \leq \frac{\varepsilon}{1-\varepsilon}\text{OPT}_{L(c)} \leq \text{OPT}_{L(c)}/3$.

Using these inequalities we also get

$$\begin{aligned}
\mathrm{dist}(S^*(c), L(c)) &\leq \mathrm{dist}(S^*(c), c) + \mathrm{dist}(c, L(c)) \\
&\leq 2\mathrm{dist}(c, \mathrm{OPT}(c)) + 2\mathrm{dist}(c, L(c)) \qquad \text{(using Inequality (2))} \\
&\leq 2\mathrm{dist}(c, \mathrm{OPT}(L(c))) + 2\mathrm{dist}(c, L(c)) \\
&\leq 4\mathrm{dist}(c, L(c)) + 2\mathrm{dist}(L(c), \mathrm{OPT}(L(c))) \\
&\leq \left(2 + \frac{4\varepsilon}{1-\varepsilon}\right) \mathrm{OPT}_{L(c)},
\end{aligned}$$

which is smaller than $3\mathrm{OPT}_{L(c)}$ for any $\varepsilon \leq 1/5$. Hence we have $c, S^*(c) \in \beta(L(c), 3\mathrm{OPT}_{L(c)})$. Since $L(c)$ is not badly cut, c and $S^*(c)$ are cut at level at most $\log(3\mathrm{OPT}_{L(c)}) + \tau(\varepsilon, d)$. Since $\mathrm{dist}(L(c), \mathrm{OPT}(L(c))) \leq \mathrm{dist}(L(c), \mathrm{OPT}(c)) \leq \mathrm{dist}(L(c), c) + \mathrm{dist}(c, \mathrm{OPT}(c)) \leq (1+\varepsilon)\mathrm{OPT}_c$, we have that $\log(3\mathrm{OPT}_{L(c)}) + \tau(\varepsilon, d) \leq \log(4\mathrm{OPT}_{L(c)}) + \tau(\varepsilon, d)$.

2. If $\mathrm{OPT}_c \leq L_c/\varepsilon$, then $2\mathrm{OPT}_c + L_c \leq 3L_c/\varepsilon$ and since c is not badly cut, the ball $\beta(c, 2\mathrm{OPT}_c + L_c)$ is cut at level at most $\log(3L_c/\varepsilon) + \tau(\varepsilon, d)$. Moreover, $S^*(c)$ lies in this ball by Inequality (2).

In all cases, c and $S^*(c)$ are separated at level at most $\log(3L_c/\varepsilon + 4\mathrm{OPT}(c)) + \tau(\varepsilon, d)$, which concludes the lemma. □

Equipped with these two lemmas, we can prove the following lemma, which concludes the section. Note again, that the bounds are for OPT defined after removing some centers in Step 1.

Lemma 4.6. *Condition on $\mathcal{I}_\mathcal{D}$ having small distortion. There exists a portal-respecting solution S in $\mathcal{I}_\mathcal{D}$ such that* $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S) + b(S) \leq (1 + O(\varepsilon))\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon \mathrm{cost}_\mathcal{I}(L))$.

Proof. The proof follows exactly the one of Lemma 3.3, making S^* portal-respecting, and using Lemma 4.2 and Lemma 4.5 to prove that the resulting solution S in $\mathcal{I}_\mathcal{D}$ has $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S) + b(S) \leq (1 + O(\varepsilon))\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon \mathrm{cost}_\mathcal{I}(L))$. □

Extension to k-Means The adaptation to k-Means – and more generally k-Clustering – can be essentially captured by the following inequality: $(x+y)^p \leq 2^p(x^p + y^p)$. Indeed, taking the example of Claim 4.4, the detour $\mathrm{dist}(c, f') \leq 3\mathrm{dist}(c, f) + 2\mathrm{dist}(c, l)$ gives a cost $\mathrm{dist}(c, f')^p = O(\mathrm{dist}(c, f)^p + \mathrm{dist}(c, l)^p)$. This follows through all the other lemmas, and therefore the above lemmas also hold for k-Means with larger constants.

4.4 The Algorithm

The algorithm follows the lines of the one for Facility Location, in Section 3.3. It first computes a constant-factor approximation L, then the hierarchical decomposition $\mathcal{D}$ (with parameter $\rho = \varepsilon 2^{-\tau(\varepsilon,d)}$) and constructs instance $\mathcal{I}_{\mathcal{D}}$. A dynamic program is then used to solve efficiently the problem, providing a solution S of cost at most $(1+\varepsilon)\text{cost}_{\mathcal{I}}(\text{OPT})$ – conditioned on the event that the instance $\mathcal{I}_{\mathcal{D}}$ has small distortion.

Dynamic programming. The algorithm proceeds bottom up along the levels of the decomposition. We give an overview of the dynamic program which is a slightly refined version of the one presented for Facility Location in Section 3.3. We make use of two additional ideas.

To avoid the dependency on k we proceed as follows. In the standard approach, a cell of the dynamic program is defined by a part of the decomposition $\mathcal{D}$, the portal parameters (as defined in Section 3.3), and a value $k_0 \in [k]$. The value of an entry in the table is then the cost of the best solution that uses k_0 centers, given the portal parameters.

For our dynamic program for the k-Median and k-Means problems, we define a cell of the dynamic program by a part B, the portal parameters $\langle \ell_1, \ldots, \ell_{n_p} \rangle$ and $\langle s_1, \ldots, s_{n_p} \rangle$ and a value c_0 in $[\text{cost}(L)/n; (1+\varepsilon)\text{cost}(L)]$. The entry of the cell is equal to the minimum number k_0 of centers that need to be placed in part B in order to achieve cost at most c_0, given the portal parameters. Moreover, we only consider values for c_0 that are powers of $(1+\varepsilon/\log n)$. The output of the algorithm is the minimum value c_0 such that the root cell has value at most k (i.e., the minimum value such that at most k centers are needed to achieve it).

The DP table can be computed the following way. For the parts that have no descendant, namely the base cases, computing the best clustering given a set of parameters can be done easily: there is at most one client in the part, and verifying that the parameter values for the centers inside the part are consistent can be done easily. At a higher level of the decomposition, a solution is obtained by going over all the sets of parameter values for all the children parts. It is immediate to see whether sets of parameter values for the children can lead to a consistent solution (similar to [25, 4]). Since there are at most $2^{O(d)}$ children parts, this gives a running time of $q^{2^{O(d)}}$, where q is the total number of parameter values.

This strategy would lead to a running time of $f(\varepsilon, d) n \log^{2^{O(d)}} n$. We can however treat the children in order, instead of naively testing all parameter values for them. We use a classical transformation of the dynamic program, in which the first table is filled using an auxiliary dynamic program. A cell of this auxiliary DP is a value c_0 in $[\text{cost}(L)/n; (1+\varepsilon)\text{cost}(L)]$, a part C, one of its children C_i, and the portal parameters for the portals of C and all its children before C_i in the given order. The entry of the cell is equal to the minimum number of centers k_0 that need to be placed in the children parts following C_i to achieve a cost of c_0 given the portal parameters. To fill this table, one can try all possible

sets of parameters for the following children, see whether they can lead to a consistent solution, and compute the minimum value among them.

Analysis – proof of Theorem 1.1 and Theorem 1.2. We first show that the solution computed by the algorithm gives a $(1+O(\varepsilon))$-approximation, and then prove the claim on the complexity.

Lemma 4.7. *Let S^* be the solution computed by the algorithm. With probability $1-2\varepsilon$, it holds that* $\text{cost}_{\mathcal{I}}(S^*) = (1+O(\varepsilon))\text{cost}_{\mathcal{I}}(\text{OPT}) + O(\varepsilon \text{cost}_{\mathcal{I}}(L))$

Proof. With probability $1-2\varepsilon$, $\mathcal{I}_{\mathcal{D}}$ has small distortion (Lemma 4.2). Following Lemma 4.6, let S be a portal-respecting solution such that $\text{cost}_{\mathcal{I}}(S) + B(S) \leq (1+O(\varepsilon))\text{cost}_{\mathcal{I}}(\text{OPT}) + O(\varepsilon \text{cost}_{\mathcal{I}}(L))$.

As in Lemma 3.4, S can be adapted to a configuration of the DP with a small extra cost. The cost incurred to the rounding of distances can be charged either to $B(S)$ or is a $O(\varepsilon)\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S)$, as in Lemma 3.4. The cost to round the value c_0 is a $(1+\varepsilon/\log n)$ factor at every level of the decomposition. Since there are $O(\log n)$ of them, the total factor is $(1+\varepsilon/\log n)^{O(\log n)} = 1+O(\varepsilon)$. Hence, we have the following:

$$\begin{aligned}
\text{cost}_{\mathcal{I}}(S^*) &= (1+O(\varepsilon))\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S^*) && (\text{as } \mathcal{I}_{\mathcal{D}} \text{ has small distortion}) \\
&= (1+O(\varepsilon))(\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) + B(S)) && (\text{by previous paragraph}) \\
&\leq (1+O(\varepsilon))\text{cost}_{\mathcal{I}}(\text{OPT}) + O(\varepsilon \text{cost}_{\mathcal{I}}(L)) && (\text{by definition of } S) \ \square
\end{aligned}$$

Lemma 4.8. *The running time of the DP is $2^{(1/\varepsilon)^{O(d^2)}} \cdot n \log^4 n$.*

Proof. The number of cells in the auxiliary DP is given by the number of parts ($O(n)$) , the number of children of a part ($2^{O(d)}$), the number of portal parameters ($(1/\varepsilon)^{2^{O(d^2)}/\varepsilon}$) and the possible values for c_0 ($O(\log^2 n)$): it is therefore $n \cdot 2^{O(d)} \cdot (1/\varepsilon)^{2^{O(d^2)}/\varepsilon} \cdot \log^2 n$.

The complexity to fill the table adds a factor $(1/\varepsilon)^{2^{O(d^2)}/\varepsilon} \cdot \log^2 n$, to try all possible combination of portal parameters and value of c_0 . Hence, the overall running time of the DP is $n \cdot (1/\varepsilon)^{2^{O(d^2)}/\varepsilon} \cdot \log^4 n = 2^{(1/\varepsilon)^{O(d^2)}} \cdot n \log^4 n$. $\square$

The proof of Theorem 1.1 and Theorem 1.2 are completed by the following lemma, which bounds the running time of the preprocessing steps.

Lemma 4.9. *For k-Median and k-Means, the total running time of the algorithms are respectively $2^{O(d)} n \log^9 n + 2^{(1/\varepsilon)^{O(d^2)}} n \log^4 n$ and $2^{O(d)} n \log^9 n + 2^{(1/\varepsilon)^{O(d^2)}} n \log^5 n$*

Proof. We need to bound the running time of three steps: computing an approximation, computing the hierarchical decomposition, and running the dynamic program.

For k-Median, a constant-factor approximation can be computed in $O(m \log^9 n) = 2^{O(d)} n \log^9 n$ time with Thorup's algorithm [38]. The split-tree decomposition can be found in $2^{O(d)} n \log n$ time as explained in Section 2. Moreover, as explained in Lemma 4.8, the dynamic program runs in time $2^{(1/\varepsilon)^{O(d^2)}} n \log^4 n$, ending the proof of the Theorem 1.1.

Another step is required for k-Means. It is indeed not known how to find a constant-factor approximation in near-linear time. However, one can notice that a c-approximation for k-Median is an nc-approximation for k-Means, using the Cauchy-Schwarz inequality. Moreover, notice that starting from a solution S, our algorithm finds a solution with cost $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(S)$ in time $2^{(1/\varepsilon)^{O(d^2)}} n \log^4 n$, as for k-Median.

Repeating this algorithm N times, using in step $i+1$ the solution given at step i, gives thus a solution of cost $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon^N)\text{cost}(S)$. Starting with $\text{cost}(S) = O(n)\text{cost}(\text{OPT})$ and taking $N = O(\log n)$ ensures to find a solution for k-Means with cost $(1 + O(\varepsilon))\text{cost}(\text{OPT})$. The complexity for k-Means is therefore the same as for k-Median, with an additional $\log n$ factor for the dynamic program term. This concludes the proof of Theorem 1.2. □

5 Other Applications of the Framework

Our techniques can be generalized to variants of the clustering problems where *outliers* are taken into account. We consider here two of them: k-Median with Outliers and its Lagrangian relaxation, Prize-Collecting k-Median. It can also be used to find a bicreteria approximation to k-Center.

5.1 Prize-Collecting k-Median

In the "prize-collecting" version of the problems, it is possible not to serve a client c by paying a penalty π_c (these problems are also called clustering "with penalties"). For a solution S, we call an *outlier for* S a client that is not served by S. Formally, an instance is a quintuple $(C, F, \text{dist}, \pi, k)$ where $(C \cup F, \text{dist})$ is a metric, k is an integer and $\pi : C \to \mathbb{R}^+$ the penalty function, and the goal is to find $S = (S_F, S_O)$ with $S_F \subseteq F$ and $S_O \subseteq C$ such that $|S_F| = k$ and $\text{cost}(S_F, S_O) = \sum_{c \in C - S_O} \text{dist}(c, S_F) + \sum_{c \in S_O} \pi_c$ is minimized. $\text{cost}(S_F)$ denotes the cost of solution S_F with the best choice of outliers (which is easy to determine)

Looking at the Prize-Collecting k-Median problem, we aim at applying the framework from Section 4. Let $L = (L_C, L_O)$ be an approximate solution. We define *badly cut* for outliers as we did for centers: an outlier c of L_O is *badly cut w.r.t.* $\mathcal{D}$ if the ball $\beta(v, 3\text{OPT}_c/)$ is cut at some level j greater than $i + \tau(\varepsilon, d)$, where OPT_c is the distance from c to the closest facility of the optimum solution OPT. Hence, Lemma 2.5 extends directly, and the probability that an outlier in L_O is badly cut is $\kappa(\varepsilon, p)$.

We now turn to the previous framework, showing how to construct a near-optimal solution containing all badly-cut centers of L. For that we transfer the definitions of the

mappings L_C, ψ (L_C maps a client to its closest center of L, and $\psi(\ell) = \{f \in \text{OPT} \mid L(f) = \ell\}$) and of the sets $L^0, L^1, L^{\geq 2}, \text{OPT}^1$, and $\text{OPT}^{\geq 2}$. We will show that this framework, with only a few modifications, leads to an approximation scheme for Prize-Collecting k-Median. Let $T = \text{OPT}$. As in Section 4, we start by removing a few centers from the optimal solution, without increasing the cost too much:

- **Step 1.** Among the facilities of $\text{OPT}^{\geq 2}$ that are not the closest of their corresponding facility in $L^{\geq 2}$, remove from T the subset $\widehat{\mathcal{H}}$ of size $\lfloor \varepsilon \cdot |\text{OPT}^{\geq 2}|/2 \rfloor$ that yields the smallest cost increase, i.e. the smallest value of $\sum_{c \in C - L_O : \text{OPT}(c) \in \widehat{\mathcal{H}}} \text{dist}(c, T - \widehat{\mathcal{H}}) + \sum_{c \in L_O : \text{OPT}(c) \in \widehat{\mathcal{H}}} \pi_c$.

The function minimized by $\widehat{\mathcal{H}}$ corresponds to redirecting all clients served in the local solution to a center of $T - \widehat{\mathcal{H}}$ and paying the penalty for clients $c \in L_O$ such that $\text{OPT}(c) \in \widehat{\mathcal{H}}$. Those clients are thus considered as outliers in the constructed solution.

Lemma 5.1. *After step 1, T has cost* $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(L)$

Proof sketch. The proof is essentially the same as Lemma 4.1, with an averaging argument: the only difference comes from the cost of removing a center from T. For any client c, the cost of removing $\text{OPT}(c)$ from T is $O(\text{OPT}_c + L_c)$: if $c \notin L^0$, the argument is the same as in Lemma 4.1, and if $c \in L_O$ the cost is π_c, which is what c pays in L. Hence the proof follows. □

Again, we denote now by OPT this solution T and define the instance $\mathcal{I}_\mathcal{D}$ according to this solution. Recall that $B_\mathcal{D}$ is the set of badly cut centers of L_C, and denote $O_\mathcal{D}$ the set of badly cut outliers of L. We say that an instance $\mathcal{I}_\mathcal{D}$ has *small distortion* if $\nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon\text{cost}(L)$ and there exists a solution S that contains $B_\mathcal{D}$ as centers and $O_\mathcal{D}$ as outliers with $\text{cost}_\mathcal{I}(S) \leq (1 + \varepsilon)\text{cost}_\mathcal{I}(\text{OPT}) + \varepsilon\text{cost}_\mathcal{I}(L)$.

To deal with the badly cut centers, there is only one hurdle to be able to apply the proof of Lemma 4.6. Indeed, when a center of OPT that serves a client c is deleted during the construction, the cost of reassigning c is bounded in Lemma 4.6 by $\text{dist}(c, S)$. However this is not possible to do when c is an outlier for S: there is no control on the cost $\text{dist}(c, S)$, and hence one has to pay the penalty π_c. It is thus necessary to find a mechanism that ensures to pay this penalty only with a probability ε for each client c. Similar to Section 4, this is achieved with the following three steps:

- **Step 2.** For each badly-cut facility $f \in L$ for which $\psi(f) \neq \emptyset$, let $f' \in \psi(f)$ be the closest to f. Replace f' by f in T^*. For all clients $c \in L_O$ such that $\text{OPT}(c) = f'$, add c as outliers.
- **Step 3.** Add all badly cut facility f' of L^0 to T^*
- **Step 4.** Add all badly cut outliers of L to the outliers of T^*.

We show next that T^* satisfies the conditions for $\mathcal{I}_\mathcal{D}$ to have small distortion with good probability.

Lemma 5.2. *The probability that $\mathcal{I}_\mathcal{D}$ has small distortion is at least $1-\varepsilon$.*

Proof. When bounding the cost increase due to Step 2, it is necessary to add as outliers all clients served by f' that are outliers in L. Since f' is deleted from T^* with probability $\kappa(\varepsilon,p)$, the expected cost due to this is $\sum_{c\in L_O}\kappa(\varepsilon,p)\cdot\pi_c \le \kappa(\varepsilon,p)\text{cost}_\mathcal{I}(L)$. Using Markov's inequality, this is at most $(\varepsilon/3)\text{cost}_\mathcal{I}(L)$ with probability $1-\varepsilon/3$.

tep 3 does not involve outliers at all. Hence, Claim 4.3 and 4.4 are still valid. Combined with the previous observation about Step 2, this proves that after Step 3, T^* contains at most k centers — including the ones in $B_\mathcal{D}$ — and has cost at most $(1+\varepsilon)\text{cost}_\mathcal{I}(\text{OPT}) + (\varepsilon/3)\text{cost}_\mathcal{I}(L)$ with probability at least $1-\varepsilon/3$.

Step 4 implies that all outliers in O_D are also outliers in the constructed solution. Moreover, since an outlier of L is badly cut with probability $\kappa(\varepsilon,p)$, the expected cost increase due to this step is at most $\kappa(\varepsilon,p)\text{cost}_\mathcal{I}(L)$. Using again Markov's inequality, this cost is at most $(\varepsilon/3)\text{cost}_\mathcal{I}(L)$ with probability $1-\varepsilon/3$.

By union-bound, the solution T^* has cost at most $(1+\varepsilon)\text{cost}_\mathcal{I}(\text{OPT}) + \varepsilon\text{cost}_\mathcal{I}(L)$ with probability $1-\varepsilon$. Hence, $\mathcal{I}_\mathcal{D}$ has small distortion with probability $1-\varepsilon$. □

Given an instance with low distortion, it is again possible to prove that there exists a near optimal portal-respecting solution, and the same DP as for k-Median can find it.

Therefore, using the polynomial time algorithm of Charikar et al. [10] to compute a constant-factor approximation, the algorithm presented in Section 4 can be straightforwardly adapted, concluding the proof of Theorem 1.4.

5.2 k-Median with Outliers

In the k-Median with Outliers problem, the number of outliers allowed is bounded by some given integer z. We do not manage to respect this bound together with having at most k facilities and a near-optimal solution: we need to relax it a little bit, and achieve a bicriteria approximation, with k facilities and $(1+O(\varepsilon))z$ outliers. For this, our framework applies nearly without a change.

The first step in the previous construction does not apply directly: the "cost" of removing a center is not well defined. In order to fix this part, Step 1 is randomized: among the facilities of $\text{OPT}^{\ge 2}$ that are not the closest of their corresponding facility in $L^{\ge 2}$, remove from T^* a random subset $\widehat{\mathcal{H}}$ of size $\lfloor \varepsilon \cdot |\text{OPT}^{\ge 2}|/2 \rfloor$.

Lemma 5.3. *After the randomized Step 1, T^* has expected cost* $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(L)$

Proof. Since there are at least $|\text{OPT}^{\geq 2}|/2$ facilities of $\text{OPT}^{\geq 2}$ that are not the closest of their corresponding facility in $L^{\geq 2}$, the probability to remove one of them is $O(\varepsilon)$. Hence, every outlier of L that is served in OPT must be added as an outlier in T^* with probability $O(\varepsilon)$ – when its serving center in OPT is deleted. Hence, the expected number of outliers added is $O(\varepsilon z)$.

Moreover, the proof of Lemma 4.1 shows that the sum of the cost of deleting all possible facilities is at most $O(\text{cost}(\text{OPT}) + \text{cost}(L))$ (adding a point as outlier whenever it is necessary). Removing each one of them with probability $O(\varepsilon)$ ensures that the expected cost of T^* after step 1 is $(1 + O(\varepsilon))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(L)$. □

The three following steps are the same as in the previous section, and the proof follows: with constant probability, the instance $\mathcal{I}_{\mathcal{D}}$ has small distortion (defined as for k-Median with penalties), and one can use a dynamic program to solve the problem on it. The DP is very similar to the one for k-Median. The only difference is the addition of a number x to each table entry, which is a power of $(1+\varepsilon/\log(n/\varepsilon))$, and represents the (rounded) number of outliers allowed in the subproblem. This adds a factor $\log^2(n/\varepsilon)/\varepsilon$ to the complexity.

It is possible to compute a constant factor approximation S in polynomial time (using Krishnaswamy et al. [26]). Hence, this algorithm is a polynomial time bicriteria approximation scheme for k-Median with outliers. As in Section 4, this directly extends to k-Means with outliers.

This concludes the proof of Theorem 1.5.

5.3 k-Center

In the k-Center problem, the goal is to place k centers such as to minimize the largest distance from a point to its serving center. We propose a bicriteria approximation, allowing the algorithm to open $(1 + O(\varepsilon))k$ centers.

For this, we change slightly the definition of badly-cut. Given a solution L with cost γ and a hierarchical decomposition $\mathcal{D}$, a center f of L is *badly cut w.r.t* $\mathcal{D}$ if the ball $\beta(f, 2^i)$ is cut at some level j greater than $i + \tau(\varepsilon, d)$, for i such that $2^{i-1} \leq 2\gamma < 2^i$.

Note that Lemma 2.5 still holds with this definition : a center f is badly cut with probability at most $\kappa(\varepsilon, p)$. Let $B_{\mathcal{D}}$ be the set of badly cut centers. We assume in the following that L is a 2-approximation, i.e. $\gamma \leq 2\text{OPT}$.

We make the crucial following observation, using the doubling property of the metric. Let f be a center of L. By definition of doubling dimension, the ball $\beta(f, \gamma)$ can be covered by 2^d balls of radius $\gamma/2 \leq \text{OPT}$. Let $\mathcal{C}_c$ be the set of centers of such balls, such that $\beta(f, \gamma) \subseteq \bigcup_{f' \in \mathcal{C}_c} \beta(f', \gamma/2)$.

Given an instance $\mathcal{I}$, we construct $\mathcal{I}_{\mathcal{D}}$ the following way: for each badly cut facility f, force all the facilities in $\mathcal{C}_f$ to be opened in any solution on $\mathcal{I}_{\mathcal{D}}$, and remove all the clients in $\beta(f, \gamma)$ from the instance. We let $\mathcal{C} = \bigcup_{f \text{ badly cut}} \mathcal{C}_f$. The structural lemma of this section is the following:

Lemma 5.4. *It holds that for all solution S of $\mathcal{I}_{\mathcal{D}}$:*

- $\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) \leq \text{cost}_{\mathcal{I}}(S)$
- $\text{cost}_{\mathcal{I}}(S \cup \mathcal{C}) \leq \max(\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S), \text{OPT})$

Proof. Since the instance $\mathcal{I}_{\mathcal{D}}$ contains a subset of clients of $\mathcal{I}$, it holds that $\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) \leq \text{cost}_{\mathcal{I}}(S)$.

Let S be a solution in $\mathcal{I}_{\mathcal{D}}$. It serves all client in $\mathcal{I}$ but the one removed: these ones are served by $\mathcal{C}$ at a cost $\gamma/2 \leq \text{OPT}$. Hence, the cost of $S \cup \mathcal{C}$ is at most $\max(\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S), \text{OPT})$. □

We now show, in a similar fashion as Lemma 3.2, that the clients in $\mathcal{I}_{\mathcal{D}}$ are cut from their serving facility of OPT at a controlled level. Recall that OPT is defined for instance $\mathcal{I}$.

Lemma 5.5. *Let c be a client in $\mathcal{I}_{\mathcal{D}}$ and $\text{OPT}(c)$ its serving facility in* OPT*. C and $\text{OPT}(c)$ are cut at level at most $\log(2\gamma) + \tau(\varepsilon, d)$.*

Proof. Let c be a client, $L(c)$ its serving center in L and $\text{OPT}(c)$ its serving center in OPT. If c is still a client in $\mathcal{I}_{\mathcal{D}}$, it means that $L(c)$ is not badly cut. Observe that $\text{dist}(L(c), \text{OPT}(c)) \leq \text{dist}(c, L(c)) + \text{dist}(c, \text{OPT}(c)) \leq \gamma + \text{OPT} \leq 2\gamma$

Let i such that $2^{i-1} \leq 2\gamma \leq 2^i$. Since $L(c)$ is not badly cut, the ball $\beta(L(c), 2^i)$ is not badly cut neither: hence, c and $\text{OPT}(c)$ (that are in this ball) are cut at level at most $i + \tau(\varepsilon, d) \leq \log(2\gamma) + \tau(\varepsilon, d)$. □

This lemma is stronger than Lemma 3.2 and 4.5: it allows us to consider only levels of the decomposition with diameter less than $2^{1+\tau(\varepsilon,d)}\gamma$.

Since the set $\mathcal{C}$ has expected size $\kappa(\varepsilon, p)k$, Markov's inequality ensures that with probability $1 - \varepsilon$ this set has size $O(\varepsilon)k$. If every part with diameter D of the hierarchical decomposition is equipped with a ρD-net (for $\rho = \varepsilon 2^{-\tau(\varepsilon,d)}$), Lemma 5.5 ensure that there exists a portal-respecting solution S with cost $\text{cost}_{\mathcal{I}_{\mathcal{D}}}(S) \leq \text{OPT} + O(\varepsilon)\gamma = (1 + O(\varepsilon))\text{OPT}$. Lemma 5.4 ensures that lifting this solution back to $\mathcal{I}$ and adding $\mathcal{C}$ as centers gives a near-optimal solution.

Using the same algorithm as for k-Median to compute a good portal-respecting solution, and computing a 2-approximation with a simple greedy algorithm (see e.g. [17]), that runs in time $O(n \log k)$ concludes the proof of Theorem 1.6.

References

[1] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 61–72. Ieee, 2017.

[2] Sanjeev Arora. Nearly linear time approximation schemes for euclidean tsp and other geometric problems. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 554–563. IEEE, 1997.

[3] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

[4] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 106–113, New York, NY, USA, 1998. ACM.

[5] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. In *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, pages 754–767, 2015.

[6] Yair Bartal and Lee-Ad Gottlieb. A linear time approximation scheme for euclidean TSP. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2013.

[7] Yair Bartal, Lee-Ad Gottlieb, Tsvi Kopelowitz, Moshe Lewenstein, and Liam Roditty. Fast, precise and dynamic distance queries. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 840–853. Society for Industrial and Applied Mathematics, 2011.

[8] Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Srinivasan Aravind, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 737–756, 2015.

[9] TH Hubert Chan, Shuguang Hu, and Shaofeng H-C Jiang. A ptas for the steiner forest problem in doubling metrics. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 810–819. IEEE, 2016.

[10] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 642–651, 2001.

[11] Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. *CoRR*, abs/1812.07721, 2018.

[12] Vincent Cohen-Addad. A fast approximation scheme for low-dimensional k-means. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 430–440, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics.

[13] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 353–364, 2016.

[14] Vincent Cohen-Addad and Chris Schwiegelshohn. On the local structure of stable clustering instances. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 49–60, 2017.

[15] Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 574–583. ACM, 2006.

[16] Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. *IEEE Transactions on Information Theory*, 55(7):3229–3242, 2009.

[17] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 434–444. ACM, 1988.

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

[19] Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour. Local search yields a ptas for k-means in doubling metrics. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 365–374. IEEE, 2016.

[20] Lee-Ad Gottlieb. A light metric spanner. In *Symposium on Foundations of Computer Science, FOCS*, 2015.

[21] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '03, 2003.

[22] Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

[23] Lingxiao Huang, Shaofeng H-C Jiang, Jian Li, and Xuan Wu. ε-coresets for clustering (with outliers) in doubling metrics. *Proceedings of FOCS 2018*.

[24] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[25] Stavros G Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. *SIAM Journal on Computing*, 37(3):757–782, 2007.

[26] Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 646–659. ACM, 2018.

[27] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.*, 222:45–58, 2013.

[28] Chen Liao and Shiyan Hu. Polynomial time approximation schemes for minimum disk cover problems. *Journal of combinatorial optimization*, 20(4):399–412, 2010.

[29] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar k-means problem is np-hard. *Theor. Comput. Sci.*, 442:13–21, 2012.

[30] Dániel Marx and Michał Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. In *Algorithms-ESA 2015*, pages 865–877. Springer, 2015.

[31] Shigeru Masuyama, Toshihide Ibaraki, and Toshiharu Hasegawa. The computational complexity of the m-center problems on the plane. *IEICE TRANSACTIONS (1976-1990)*, 64(2):57–64, 1981.

[32] Jirı Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

[33] Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13.

[34] Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

[35] Adam Meyerson. Online facility location. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 426–431. IEEE, 2001.

[36] Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on computing*, 28(4):1298–1309, 1999.

[37] K. Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 281–290. ACM, 2004.

[38] Mikkel Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.

A Appendix

A.1 Proofs for Section 2

Proof of Lemma 2.3. We present the algorithm constructing the hierarchical decomposition, and prove the lemma as a second step.

Without loss of generality, assume that the smallest distance in the metric is 1: the aspect-ratio Δ is therefore the diameter of the metric. Start from a hierarchy of nets $Y_0 := V, \ldots, Y_{\log(\Delta)}$ such that Y_i is a 2^{i-2}-net of Y_{i-1}. Moreover, pick a random order on the points V and a random number $\tau \in [1/2, 1)$. The hierarchical decomposition $\mathcal{D}$ is defined inductively, starting from $\mathcal{B}_{\lceil \log \Delta \rceil} = V$. To partition a part B at level i into subparts at level $i-1$, do the following: for each $y \in Y_{i-1} \cap B$ taken in the random order, define $B \cap \beta(y, \tau 2^i)$ to be a part at level $i-1$ and remove $B \cap \beta(y, \tau 2^i)$ from B.

When we assume access to the distances through an oracle, it is possible to construct this hierarchy and augment it with the set of portals in time $(1/\rho)^{O(d)} n \log(\Delta)$. Moreover, these portals can be made *nested*, meaning that portals at level $i+1$ are also portals at level i [22, 15].

We prove now that this hierarchical decomposition has the required properties. The diameter of each part is bounded by 2^{i+1} by construction; therefore to have Property 2 it is enough to make $\mathcal{P}_i$ an $(\rho 2^{i+1})$-net of V. Property 2.1 ensures the conciseness, and the definition of a net ensures that every point is at distance $\rho 2^{i+1}$ of $\mathcal{P}_i$, which implies the preciseness. The construction ensures that a part B at level $i-1$ is split in at most $|B \cap Y_{i-1}|$ parts of level i, which is $2^{O(d)}$ following Property 2.1. Proving the scaling property requires a bit more work.

The two ingredients needed for this part stem from the construction of the decomposition: the diameter of any part at level i is at most 2^{i+1}, and the minimum distance between two points of Y_i is bigger than 2^{i-2}.

These two properties are enough in order to prove our lemma. Let i be a level such that $2^i \leq r$: then $r/2^i = \Omega(1)$ so there is nothing to prove. Otherwise, we proceed in two steps. First, let us count the number of level i parts that could possibly cut a ball $\beta(x, r)$. A level i part is included in a ball $\beta(y, 2^i)$ for some $y \in Y_i$; therefore if $\mathrm{dist}(x, y) > r + 2^i$ then y's part cannot cut $\beta(x, r)$. So it is required that $\mathrm{dist}(x, y) \leq r + 2^i \leq 2 \cdot 2^i$. But since the minimum distance between two points of Y_i is 2^{i-2}, and Y_i has doubling dimension d, we have $|Y_i \cap \beta(x, 2 \cdot 2^i)| = 2^{d \log(2^i/2^{i-2})} = 2^{2d}$. Thus there is only a bounded number of parts to consider.

We prove for each of them that the probability that it cuts $\beta(x, r)$ is $O(r/2^i)$. A union-bound on all the possible parts is then enough to conclude. Let therefore $y \in Y_i \cap \beta(x, 2 \cdot 2^i)$, and x_m and x_M be the respective closest and farthest point of $\beta(x, r)$ from y. A necessary condition for y's part to cut $\beta(x, r)$ is that the diameter of the part is in the open interval $(d(y, x_m), d(y, x_M))$. Since $x_m, x_M \in \beta(x, r)$ this interval has size $2r$, and the radius of the part is picked uniformly in $[2^i/2, 2^i)$. Therefore the probability that the radius of the part falls in $(d(y, x_m), d(y, x_M))$ is at most $4r/2^i$. And finally, the probability that y's part cuts $\beta(x, r)$ is indeed $4r/2^i$.

By a union-bound over all the parts that could possibly cut $\beta(x, r)$ we obtain the claimed probability $\Pr[\mathcal{C} \text{ cuts } \beta(x, r) \text{ at a level } i] = 2^{2d+2} r/2^i$. □

Lemma A.1. *Let P be a problem among Facility Location, k-Median or k-Means. Given an instance $\mathcal{I}$ on a metric (V, dist) with n points, $\varepsilon > 0$, and a constant-factor approximation for P on $\mathcal{I}$, there exists a linear-time algorithm that outputs a set of instances $\mathcal{I}_1, \ldots, \mathcal{I}_m$ on metrics $(V_1, \mathrm{dist}_1), \ldots, (V_m, \mathrm{dist}_m)$, respectively, such that*

- *$V_1, \ldots, V_m$ is a partition of V*
- *for all i, (V_i, dist_i) has aspect-ratio $O(n^5/\varepsilon)$,*
- *if (V, dist') is the metric where distances between points of the same part V_i are given by dist_i while distances between points of different parts is set to ∞, and OPT is the optimum solution to $\mathcal{I}$, then*
  - *there exists a solution on (V, dist') with cost $(1 + \varepsilon/n)\mathrm{cost}(\mathrm{OPT})$, and*
  - *any solution on (V, dist') of cost X induces a solution of cost at most $X + \varepsilon \mathrm{cost}(\mathrm{OPT})/n$ in $\mathcal{I}$.*

Proof. The cost of the constant-factor approximation is an estimate γ on the cost of the optimum solution OPT: $\gamma = \Theta(\mathrm{cost}(\mathrm{OPT}))$. It is then possible to replace all distances longer than 2γ by ∞: distances longer than γ will indeed never be used by solution with cost better than γ, so the cost of these solutions is preserved after this transformation. The

distance matrix do not respect the triangle inequality anymore: thus we replace it with its metric closure. We say that two vertices are *connected* if their distance is not ∞, and call a connected component any maximal set of connected vertices. The transformation ensured that any connected component has diameter at most $2n\mathrm{OPT}$, and that every cluster of OPT is contained inside a single connected component. Moreover, any connected component has doubling dimension $2d$: indeed, a subspace of a metric with doubling dimension d has a doubling dimension at most $2d$. Note also that this transformation can be made implicitly: every time the algorithm queries an edge, it can replace the result by ∞ if necessary.

To identify the connected component, the algorithm builds a spanner with the algorithm of [22]: the connected components of the spanner are exactly the ones of our metric, and can be found in linear time.

Then, for each connected component, the algorithm defines an instance of the more general version of the clustering problem by the following way. It first sets $\chi(v) = 1$ for all vertex v. Then, it iterates over all edges, it contracts every edge (u, v) with length less than $(\varepsilon \cdot \gamma/n^3)$ to form a new vertex w and sets $\chi(w) = \chi(u) + \chi(v)$.

Now, we aim at reconstructing a metric from this graph. We will do it in an approximate way: for all connected points u, v of connected component i, we set $\mathrm{dist}_i(u, v)$ to be 0 if u and v are merged in the graph, and otherwise $\mathrm{dist}(u, v)$. This ensures that $\varepsilon \cdot \gamma/n^3 \leq \mathrm{dist}_i(u, v) \leq 2n\gamma$, hence the aspect-ratio of $\mathcal{I}_i$ is $O(n^5/\varepsilon)$. Moreover, every distance is preserved up to an additive $O(\varepsilon \cdot \mathrm{cost}(\mathrm{OPT})/n^2)$.

Since every cluster of OPT is contained inside a single connected component, this ensures that OPT induces a solution of cost $(1+\varepsilon/n)\mathrm{cost}(\mathrm{OPT})$ on $\bigcup \mathcal{I}_i$. Moreover, lifting a solution in $\bigcup \mathcal{I}_i$ to $\mathcal{I}$ costs at most $\varepsilon\mathrm{cost}(\mathrm{OPT})/n^2$ per pair (client, center) and therefore $\varepsilon\mathrm{cost}(\mathrm{OPT})/n$ in total. □

If the problem considered is Facility Location, it is easy to merge the solutions on subinstances: since there is no cardinality constraint, the global solution is simply the union of all the solutions. The hard constraint on k makes things a bit harder. Note that the dynamic program presented in Section 4 naturally handles it without any increase in its complexity: however, for completeness we present now a direct reduction.

Lemma A.2. *Given a problem P among k-Median or k-Means, a set of instances $(\mathcal{I}_1, \mathrm{dist}_1)$, …, $(\mathcal{I}_m, \mathrm{dist}_m)$ given by Lemma A.1 and an algorithm running in time $n_i(\log n_i)^\alpha t(\Delta)$ to solve P on instances with n_i points and aspect-ratio Δ, there exists an algorithm that runs in time $O(n(\log n)^{\alpha+2}t(O(n^4/\varepsilon)))$ to solve P on $\bigcup \mathcal{I}_i$.*

Proof. First, note that the optimal solution in $\bigcup \mathcal{I}_i$ is $O(n^5/\varepsilon)$, since the maximal distance in any of $\mathcal{I}_1, \ldots \mathcal{I}_m$ is n^4/ε. Using this fact, we build a simple dynamic program to prove the lemma. For all $i \leq m$ and $j \leq \log_{1+\varepsilon/\log n}(n^5/\varepsilon)$, let $k_{i,j}$ be the minimal k' such that the cost of P with k' centers in $\mathcal{I}_i$ is at most $(1 + \varepsilon/\log n)^j$. $k_{i,j}$ can be computed with a simple binary search, using the fact that the cost of a solution is decreasing with k'.

Given all the $k_{i,j}$, a simple dynamic program can compute $k_{\geq i,j}$, the minimal number of centers needed to have a cost at most $(1+\varepsilon)^j$ on $\mathcal{I}_i, \dots \mathcal{I}_m$ (the $\varepsilon/\log n$ becomes a simple ε because of the accumulation of errors). The solution for our problem is $(1+\varepsilon)^j$, where j is the minimal index such that $k_{\geq 1,j} \leq k$.

The complexity of computing $k_{i,j}$ is $O(\log k \cdot n_i (\log n)^\alpha t(O(n^4/\varepsilon)))$, hence the complexity of computing all the $k_{i,j}$ is $O(n(\log n)^{\alpha+2} t(O(n^4/\varepsilon))$. The complexity of the dynamic program computing $k_{\geq i,j}$ is then simply $O(m \log n) = O(n \log n)$, which concludes the proof. □

A.2 Portal Respecting Paths and Solutions

Recall that any part $B \in \mathcal{B}_i$ of the decomposition $\mathcal{D} = \{\mathcal{B}_0, \dots, \mathcal{B}_{|\mathcal{D}|}\}$ comes with a set of portals $\mathcal{P}_B$ with the properties listed in Lemma 2.3. In a portal-respecting solution, every client connects to a facility by going in and out of parts of the decomposition only at designated portals. More concretely, a *path* in a metric between two nodes u and v is given by a sequence of nodes $w_1, \dots, w_k$, where $u = w_1, v = w_k$, and its length is $\sum_{j=1}^{k-1} \operatorname{dist}(w_j, w_{j+1})$. A solution can be seen as a set of facilities, together with a path for each client that connects it to a facility, and the cost of the solution is given by the sum over all path lengths. We say a path $w_1, \dots, w_k$ is *portal-respecting* if for every pair w_j, w_{j+1}, whenever w_j and w_{j+1} lie in different parts $B, B' \in \mathcal{B}_i$ of the decomposition $\mathcal{D}$ on some level i, then these nodes are also portals at this level, i.e., $w_j, w_{j+1} \in \mathcal{P}_B \cup \mathcal{P}_{B'}$. As explained in Lemma A.3, if two vertices u and v are cut at level i, then there exists a portal-respecting path from u to v of length at most $\operatorname{dist}(u,v) + 16\rho 2^i$. We define a *portal-respecting* solution to be a solution such that each path from a client to its closest facility in the solution is portal-respecting.

Lemma A.3. *If two vertices u and v are cut by a decomposition at level i, there exists a portal-respecting path of length* $\operatorname{dist}(u,v) + 16\rho 2^i$ *that connects u to v.*

Proof. If u and v are cut on level i, then they lie in the same part $B \in \mathcal{B}_{i+1}$ on level $i+1$ of the decomposition $\mathcal{D}$. As each part on level 0 of $\mathcal{D}$ is a singleton set, both u and v are portals on that level. Now consider the path that starts in $u = w_1$, and for each $j \geq 1$ connects w_j to the closest portal $w_{j+1} \in \mathcal{P}_B$ of the part $B \in \mathcal{B}_j$ on the next level j, until level $i+1$ is reached. This yields a portal-respecting path P_u, as portals are nested, i.e., each w_j is a portal on every level less than j. A similar procedure finds a portal-respecting path P_v from the other endpoint v up to level $i+1$ through portals of levels below $i+1$. Since u and v lie in the same part on level $i+1$, we may obtain a portal-respecting path from u to v by first following P_u up to level $i+1$, then connecting to the endpoint of P_v that is a portals of level $i+1$, and then following P_v all the way down to v. The length of this portal-respecting path is at most $\operatorname{dist}(u,v) + 4\sum_{j \leq i+1} \rho 2^{j+1} = \operatorname{dist}(u,v) + O(\rho 2^i)$, due to the triangle inequality and the preciseness property of the portals (cf. Lemma 2.3). □