A nearly linear-time approximation scheme for the Euclidean k-median problem*

Stavros G. Kolliopoulos† Satish Rao‡

Abstract

In the *k-median* problem we are given a set N of n points in a metric space and a positive integer k. The objective is to locate k medians among the points so that the sum of the distances from each point in N to its closest median is minimized. The k-median problem is a well-studied, NP-hard, basic clustering problem, which is closely related to facility location. Obtaining constant-factor approximations for this problem, even for the 2-dimensional Euclidean metric, had long been an elusive goal. First Arora, Raghavan and Rao gave a randomized polynomial-time approximation scheme by extending techniques introduced originally by Arora for the Euclidean TSP. For any fixed $\varepsilon > 0$, their algorithm outputs a $(1+\varepsilon)$-approximation in $O(nkn^{O(1/\varepsilon)} \log n)$ time.

In this paper we provide a randomized approximation scheme for points in d-dimensional Euclidean space, with running time $O(2^{O((1+\log(1/\varepsilon)/\varepsilon)^{d-1})} n \log n \log k)$, which is nearly linear for any fixed ε and d. Moreover our method provides the first polynomial-time approximation scheme for k-median and uncapacitated facility location instances in d-dimensional Euclidean space for any fixed $d > 2$. To obtain the improvement we develop a structure theorem to describe hierarchical decomposition of solutions. The theorem is based on an *adaptive decomposition* scheme, which guesses at every level of the hierarchy the structure of the optimal solution and modifies accordingly the parameters of the decomposition. We believe that our methodology is of independent interest and may find applications to further geometric problems.

Keywords: approximation algorithms, approximation schemes, k-median, facility location, Euclidean space, linear time. *AMS subject classifications:* 68Q25, 90B10, 90B12, 90B35.

Running head: Nearly linear-time approximation scheme

*Part of this work was performed while the authors were at the NEC Research Institute, Inc, 4 Independence Way, Princeton, NJ 08540.

†Department of Computing and Software, Faculty of Engineering, McMaster University, Hamilton, Ontario, L8S 4K1, Canada (stavros@mcmaster.ca). Research partially supported by NSERC Grant 227809-00 and a CFI New Opportunities Award.

‡Computer Science Division, University of California Berkeley, CA 94720, USA (satishr@cs.berkeley.edu).

1 Introduction

In the *k-median* problem we are given a set N of n points in a metric space and a positive integer k. The objective is to locate k *medians* (facilities) among the points so that the sum of the distances from each point in N to its closest median is minimized. The k-median problem is a well-studied, NP-hard problem which falls into the general class of *clustering* problems: partition a set of points into clusters so that the points within a cluster are close to each other with respect to some appropriate measure. Moreover k-median is closely related to *uncapacitated facility location,* a basic problem in the operations research literature (see, e.g., [10]). In the latter problem except for the set N of points we are given also a cost c_i for *opening* a facility at point i. The objective is to open an unspecified number of facilities at a subset of N so as to minimize the sum of the cost to open the facilities (*facility cost*) plus the cost of assigning each point to the nearest open facility (*service cost*). In this paper we provide a fast approximation scheme for the problem when the input lies in a Euclidean space. A *ρ-approximation algorithm* for a minimization problem, $\rho > 1$, computes in time polynomial in the input size a feasible solution of cost at most ρ times the optimum. An *approximation scheme* computes for any fixed $\varepsilon > 0$, a $(1+\varepsilon)$-approximatte feasible solution in time polynomial in the input size and $1/\varepsilon$.

1.1 Previous Work

The succession of results for k-median is as follows. Lin and Vitter [18] used their filtering technique to obtain a solution of cost at most $(1+\varepsilon)$ times the optimum but using $(1+1/\varepsilon)(\ln n+1)k$ medians. They later refined their technique to obtain a solution of cost $2(1+\varepsilon)$ while using at most $(1+1/\varepsilon)k$ medians [17]. The first non-trivial approximation algorithm that achieves feasibility as well, i.e. uses k medians, combined the powerful randomized algorithm by Bartal for approximation of metric spaces by trees [3, 4] with an approximation algorithm by Hochbaum for k-median on trees [14]. The ratio thus achieved is $O(\log n \log\log n)$. This algorithm was subsequently refined and derandomized by Charikar et al. [6] to obtain a guarantee of $O(\log k \log\log k)$. Recently, Charikar and Guha and independently Tardos and and Shmoys reported the first constant-factor approximations [8]. In contrast, the uncapacitated facility location problem, in which there is no a priori constraint on the number of facilities, seems to be better understood. Shmoys, Tardos and Aardal [21] gave a 3.16 approximation algorithm. This was later improved by Guha and Khuller [13] to 2.408 and to 1.736 by Chudak [9]. After the first publication of this work [16] some additional results on k-median and facility location appeared in [7, 15].

In this paper we focus on the case when the underlying metric is Euclidean. Until the work of Arora, Raghavan and Rao [2], this case was not known to be any easier to approximate than a general metric. These authors gave a randomized polynomial-time approximation scheme for k-median when the points lie on the Euclidean plane [2]. For any fixed $\varepsilon > 0$, their algorithm outputs a $(1+\varepsilon)$-approximation with probability $1-o(1)$ and runs in $O(nkn^{O(1/\varepsilon)}\log n)$ time, worst case. For facility location they gave an approximation scheme with running time $O(n^{1+O(1/\varepsilon)}\log n)$. This development followed the breakthrough approximation schemes of Arora [1] for the Traveling Salesman Problem and other geometric problems. While the work in [2] used techniques from the TSP approximation scheme, the different structure of the optimal solutions for k-median and TSP necessitated the development of a new structure theorem to hierarchically decompose solutions. We elaborate further on this issue during the exposition of our results in the next paragraph. The dependence of the running time achieved by the methods of Arora, Raghavan and Rao on $1/\varepsilon$ is particularly high. For example, the approximation scheme can be extended to higher-dimension instances but runs in quasi-polynomial time $O(n^{(\log n/\varepsilon)^{d-2}})$ for a set of points in R^d with fixed

$d > 2$.

1.2 Results and Techniques

Results. We provide a randomized approximation scheme for k-median on the Euclidean plane. For any fixed $\varepsilon > 0$, our scheme outputs in expectation a $(1+\varepsilon)$-approximate solution, in time

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

worst case. Our time bound represents a drastic improvement on the result in [2]. For any fixed accuracy ε desired, the dependence of the running time on $1/\varepsilon$ translates to a (large) constant hidden in the near-linear asymptotic bound $O(n \log n \log k)$ compared to the exponent of a term polynomial in n in the bound of Arora, Raghavan and Rao. Moreover, for inputs in R^d, our algorithm extends to yield a running time of

$$O\left(2^{O\left((1+\frac{\log(1/\varepsilon)}{\varepsilon})^{d-1}\right)} n \log k \log n\right)$$

which yields for the first time a polynomial-time approximation scheme for any fixed $d > 2$. The ideas behind the new k-median algorithm yield also improved, nearly linear-time, approximation schemes for uncapacitated facility location. Our time bounds for the latter problem hold under the assumption that a polynomial in n approximation is available for the value of the service cost. An example of such a case is when all the inter-point distances are polynomially related. We now elaborate on the techniques we use to obtain our results.

Techniques. A contribution of this paper lies in the new ideas we introduce to overcome the limitations of the approach employed by Arora, Raghavan and Rao in [2]. To describe these ideas we sketch first some previous developments, starting with the breakthrough results in [1] (see also [19] for a different approximation scheme for Euclidean TSP).

A basic building block for Arora's results on TSP [1] was a structure theorem providing insight into how much the cost of an optimal tour could be affected in the following situation. Roughly speaking, the plane is recursively dissected into a collection of rectangles of geometrically decreasing area, represented by a quadtree data structure. For every box in the dissection one places a fixed number, dependent on the desired accuracy ε, of equidistant *portals* on the boundary of the box. The optimal TSP tour can cross between adjacent rectangles any number of times; a *portal-respecting* tour is allowed to cross only at portals. How bad can the cost of a deflected, portal-respecting, tour be compared to the optimum? Implicitly, Arora used a charging argument on the edges in an optimal solution to show that the edges could be made to be portal respecting. We sketch now his approach which was made explicit and applied to k-median in [2].

Given a set of open facilities a k-median solution is a set of edges assigning every point to an open facility. We can assume that the input is surrounded by a rectangle with side length polynomial in n (cf. Section 2). At level i of the dissection, the rectangles at this level with sidelength 2^i are cut by vertical and horizontal lines into rectangles of sidelength 2^{i-1}. The x- and y-coordinates of the dissection are randomly shifted at the beginning, so that the probability that an edge e in a solution is cut at level i is $O(\text{length}(e)/2^i)$. Let m denote the number of portals along the dissection lines. If e is cut at level i it must be deflected through a portal, paying additional cost $O(2^i/m)$. Summing over all the $O(\log n)$ levels of the decomposition, the expected deflection cost of edge e

in a portal-respecting solution is at most

$$\sum_{i=1}^{O(\log n)} O(\frac{\text{length}(e)}{2^i}(2^i/m)) \tag{1}$$

Selecting $m = \Theta(\log n/\varepsilon)$ and applying the dissection to the optimal solution we obtain the existence of a portal-respecting solution of cost $(1+\varepsilon)OPT$. Once the existence has been shown, dynamic programming can be used to compute the best portal-respecting solution. The running time of the dynamic programming contains a $k2^{O(m)}$ term, hence the $kn^{1/\varepsilon}$ term in the overall running time.

Arora additionally used a "patching lemma" argument to show that the TSP could be made to cross each box boundary $O(1/\epsilon)$ times. This yielded an $O(n(\log n)^{O(1/\varepsilon)})$ time algorithm for TSP. (This running time was subsequently improved by Rao and Smith to $O(2^{O(1/\varepsilon^2)} + n\log n)$ [20] while still using $\Theta(\log n/\varepsilon)$ portals). The k-median method did not, however, succumb to a patching lemma argument, thus the running time for the algorithms in [2] remained $O(kn^{O(1/\epsilon)})$.

Our method reduces the number m of portals to $O(1+\log(1/\epsilon)/\epsilon)$. *That is, we remove the* $\log n$ *factor in the number of portals that appears to be inherent in Arora, Raghavan, and Rao's charging based methods and even in Arora's charging plus patching based methods.*

Adaptive dissection. We outline some of the ideas behind the reduced value for m. The computation in (1) exploits linearity of expectation by showing that the "average" dissection line cutting an edge is short enough. The complicated dependencies among the dissection lines across all $O(\log n)$ levels seem too complicated to reason about directly. On the other hand, when summing the expectations across all levels an $O(\log n)$ factor creeps in, which apparently has to be offset by setting m to $\log n/\varepsilon$. We provide a new structure theorem to characterize the structure of near-optimal solutions. In contrast to previous approaches, given a rectangle at some level in the decomposition, it seems a good idea to choose several possible "cuts" hoping that one of them will hit a small number of segments from the optimum solution. This approach gives rise to the *adaptive dissection* idea, in which the algorithm "guesses" the structure of the part of the solution contained in a given rectangle and tunes accordingly the generation of the sub-rectangles created for the next level of the dissection. In the k-median problem the guess consists of the area of the rectangle which is empty of facilities. Let L be the maximum side length of the sub-rectangle containing facilities. Cutting close to the middle of this sub-rectangle with a line of length L should, in a probabilistic sense, mostly dissect segments from the optimal solution of length $\Omega(L)$, forcing them to deflect by $L/m = \varepsilon L$. A number of complications arise by the fact that a segment may be cut by both horizontal and vertical dissection lines. We note that the cost of "guessing" the empty area is incorporated into the size of the dynamic programming lookup table by trying all possible configurations. Given the preeminence of recursive dissection in approximation schemes for Euclidean problems ([1, 2, 20]) we believe that the adaptive dissection technique is of independent interest and may prove useful in other geometric problems as well.

Although the adaptive dissection technique succeeds in reducing the required number of portals to $\Theta(\log(1/\varepsilon)/\varepsilon)$ and thus asymptotically improve the dependence of the running time on $1/\varepsilon$, the dynamic program has still to enumerate all possible rectangles. Compared to the algorithm in [2] we apparently have to enumerate even more rectangles due to the "guess" for the areas without facilities. We further reduce the size of the lookup table by showing that the boundaries of the possible rectangles can be appropriately spaced and still capture the structure of a near-optimal solution.

The outline of the paper is as follows. In Section 2 we give definitions and preprocessing steps. In Section 3 we prove the new structure theorem and obtain the reduced number of portals. In Section

4 we provide a modified structure theorem which yields a small size for the dynamic program table. In Section 5 we present the dynamic program and some extensions of the main result.

2 Preliminaries

An *edge* (u, v) is a line segment connecting input points u and v. Given a selection of open facilities, an *assignment edge* is an edge (u, v) such that exactly one of u or v is an open facility. An *assignment* is a set E of assignment edges such that every point which does not host a facility appears exactly once as an endpoint. For minimum cost every point must of course be assigned to its closest open facility. The *sidelength* of a rectangle with sides parallel to the axes is the length of its largest side.

We assume that the input points are on a unit grid of size polynomial in the number of input points. This assumption is enforced by a preprocessing phase described in [2]; the preprocessing incurs an additive error of $O(1/n^c)$ times the optimal value, for some constant $c > 0$. The assumption is needed to ensure the depth of the recursive dissection is $O(\log n)$. Within the same additive error we can assume that no two input points lie on the same vertical grid line. The latter assumption simplifies the presentation.

In [2] it is shown that if a polynomial-factor approximation is available for the value of the service cost, i.e., a range $[D, n^c D]$ where this value must lie, then the above assumptions on the points can be enforced by a simple plane sweep. An algorithm to compute this range for the k-median problem is the minmax clustering algorithm of Gonzalez [12]. A faster algorithm for the same problem was given by Feder and Greene [11]. The latter runs in $O(n \log k)$ time on the plane. The total preprocessing time for $d = 2$ is $O(n \log n)$. For general dimension d the preprocessing time is $O(dn \log n + kd^d \log(kd^d))$ [11].

3 The Structure Theorem

In this section we prove our basic Structure Theorem that shows the existence of approximately optimal solutions with a simple structure. This theorem can yield directly a dynamic-programming algorithm whose running time dependence on ε is no worse than $2^{O(\log(1/\varepsilon)/\varepsilon)}$. Our exposition focuses on 2-dimensional Euclidean instances. It is easy to generalize to d dimensions. Given a set of points N and a set of facilities $F \subset N$, we define the *greedy cost under* F to be the cost of assigning each point to its closest facility. If F is the set of facilities open in the optimal solution, the greedy cost under F is the optimal cost.

We proceed to define a recursive randomized decomposition. The decomposition uses two processes: the Sub-Rectangle process, and the Cut-Rectangle process.

Sub-Rectangle:
Input: a rectangle B containing at least one facility.
Process: Find the minimal rectangle B' containing all the facilities. Let its maximum sidelength be s. Grow the rectangle by $s/3$ in each dimension. We call the grown rectangle B''.
Output: $B_s = B'' \cap B$.

Notice that $B - B_s$ contains no facilities. If $B_s \neq B$, we call B_s a *proper sub-rectangle.*

Cut-Rectangle:
Input: a rectangle B containing at least one facility.

Process: Randomly cut the rectangle into two rectangles with a line that is orthogonal to the middle third of the maximal side of the rectangle.
Output: The two created rectangles.

Remark 3.1 *Given as input a rectangle with constant aspect ratio, both processes output rectangles with constant aspect ratio. This remark will be useful in Section 4.*

The recursive method applies alternately the Sub-Rectangle and Cut-Rectangle processes to produce a decomposition of the original rectangle containing the input. We emphasize that in this section we do not use an apriori random shift of the coordinate system as Arora in [1]. The Sub-Rectangle process would diminish any randomization introduced at the beginning of the dissection. The randomization required by the upcoming Facts 3.1 and 3.2 is introduced by Cut-Rectangle. We also observe that the original rectangle is not necessarily covered by leaf rectangles in the decomposition, due to the sub-rectangle steps.

We place $m+1$ evenly spaced points on each side of each rectangle in the dissection, where m will be defined later and depends on the accuracy ε of the sought approximation. We call these points *portals.* We define a *portal-respecting path* between two points to be a path between the two points that only crosses rectangles that enclose one of the points at portals. We define the *portal-respecting distance* between two points to be the length of the shortest portal-respecting path between the points. We begin by giving three technical lemmata which will be of use in the main Structure Theorem. Lemma 3.1 has a straightforward proof and gives the motivation behind the decomposition. We want short assignment edges in a given solution to be separated by rectangles of small sidelength.

Lemma 3.1 *If the first rectangle R in the dissection to separate points v and w has sidelength D, the difference between the portal respecting distance and the geometric distance between v and w is $O(D/m)$.*

We define a *cutting* line segment in the decomposition to be (i) either a line segment l that is used in the Cut-Rectangle process to divide a rectangle R into two rectangles or (ii) a line segment l used to form the boundary of a proper sub-rectangle R in the Sub-Rectangle procedure. In both cases we say that *l cuts R*. We define the *sidelength* of a cutting line l as the sidelength of the rectangle cut by l. Observe that the length of a cutting line is upperbounded by its sidelength.

Lemma 3.2 *If any two parallel cutting line segments produced by the application of Cut-Rectangle are within distance L, one of the line segments has sidelength at most $3L$.*

Proof: Let l_1, l_2 be the two cutting segments at distance L. Assume without loss of generality that they are both vertical, l_1 is the longer of the two lines, l_1 is on the left of l_2 and cuts a rectangle R of sidelength greater than $3L$ into R_1 and R_2. Then l_1 is produced first in the decomposition. Thus l_2 is contained within R_2 (see Fig. 1a), and since it comes second can only cut a rectangle R_2' contained within R_2. By the definition of Cut-Rectangle, if s is the sidelength of R_2', l_2 is drawn at least $s/3$ away from the left boundary of R_2' which implies $s/3 \leq L$. Thus $s < 3L$. $\Box$

The next lemma relates the length of a cutting line segment produced by Sub-Rectangle to the length of any assignment edges it intersects. We slightly abuse terminology and say that an edge (v, f) is *separated* when a cutting line separates v and f.

Lemma 3.3 *Let f be a facility. If a cutting line segment σ produced by Sub-Rectangle intersects an assignment edge (v, f) of length D, σ has sidelength at most $5D$.*

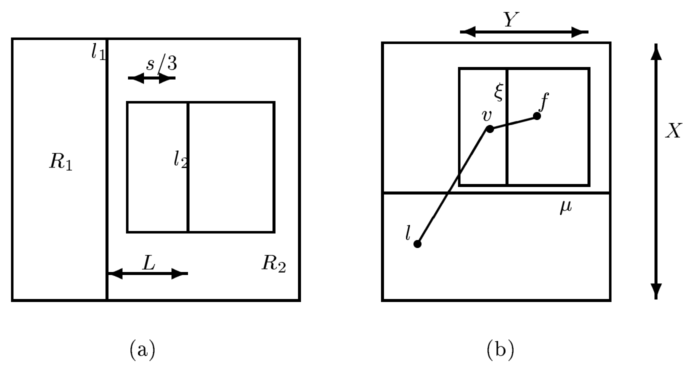

Figure 1: (a) Demonstration of Lemma 3.2. (b) Case A2 in the proof of Theorem 3.1.

Proof: Let R be the rectangle cut by σ. Observe that f must be contained in R. Let s be the sidelength of the rectangle R. Without loss of generality assume that the horizontal dimension of R is maximal. By the definition of Sub-Rectangle there is $y < s$ such that $s = (5/3)y$ and the two vertical strips of width $y/3$ at the sides of R are empty of facilities. Therefore $y/3 < D$ which implies that $s < 5D$. □

Consider the optimal solution with k given medians among the points. In the Structure Theorem we will prove the existence of a near-optimal solution in which a point v is always assigned to its closest or second-closest median. The *modified cost* of an assignment E is the sum over all the points of the portal-respecting distances to their respective assigned facility. Since the Structure Theorem assumes that the set of open facilities is given, it applies more generally to uncapacitated facility location.

We provide first two calculations that will be of use in the proof of the theorem.

Fact 3.1 *Let $\mathcal{E}(\Delta, Z)$ denote the event that an edge of length Δ is separated by a cutting line of sidelength Z that is produced by Cut-Rectangle.*

$$Pr[\mathcal{E}(\Delta, Z)] \leq 3\Delta/Z.$$

Sidelengths increase geometrically in the dissection. Therefore a consequence of Fact 3.1 is that the probability that an edge of length Δ is separated by a cutting line of sidelength Z or more that is produced by Cut-Rectangle is at most $O((\Delta/Z))$.

Fact 3.2 *Let $\mathcal{E}_{\geq}(\Delta, Z)$ denote the event that an edge of length Δ is separated by a cutting line of sidelength Z or more that is produced by Cut-Rectangle.*

$$Pr[\mathcal{E}_{\geq}(\Delta, Z)] = O(\Delta/Z).$$

Theorem 3.1 *(**Structure Theorem**) Let $m > 0$ be any integer and $F \subset N$ a set of open facilities. There is an assignment E such that the expected difference between the modified cost of E and the greedy cost C under F is $O(C \max\{1, \log(m)\}/m)$.*

Proof of theorem: By linearity of expectation, it suffices to bound the expected cost increase for a given assignment edge. For a point v we define $f \in F$ as v's closest facility and assume f to be to the right of v (without loss of generality.) We define $l \in F$ as the closest facility to the left of v. We denote the distance from v to f by D, and the distance from v to l by L. The idea behind the analysis of the portal-respecting solution is that assigning v to either f or l (a decision based on the amount by which the decomposition distorts each distance) will be enough to show near-optimal modified cost. The proof of the theorem shows how to actually construct the assignment E.

We assume without loss of generality that v and f are separated for the first time by a vertical cutting line. We can turn the configuration on its side and do the same argument if this condition does not hold.

The semicircle of diameter $2L$ centered at v and lying entirely to the left of the vertical line passing through v is empty in its interior. Therefore we obtain:

Lemma 3.4 *Let the distance from a point v to its closest open facility f be D and the distance from v to its second closest facility l be L. In the decomposition, v and f are not separated for the first time by a cutting line of sidelength in the interval $(8D, L/2)$.*

Proof of lemma: We know that v and l are separated by the time the sidelength of any enclosing rectangle is at most L. Observe that by Lemma 3.3, a cutting line of sidelength $> 5D$ separating v and f can only be produced by Cut-Rectangle. We will show that Cut-Rectangle cannot produce such a cutting line with sidelength in $(8D, L/2)$.

Without loss of generality we assume the relative positioning of v, f, l given above. For any rectangle of sidelength $L/2$ containing v, there are no facilities to the left of v. Thus, by the Sub-Rectangle process on input of sidelength $s < L/2$, the left boundary of any rectangle of sidelength $s < L/2$ that contains v is within distance at most $s/5$ of v. This is because in Sub-Rectangle the maximal rectangle empty of facilities is grown by at most a third in each direction. By the Cut-Rectangle process on input of sidelength s, any cutting line for a rectangle box is at least $s/3$ to the right of its left boundary which implies that the cutting line is at least $s/3 - s/5 = 2s/15$ to the right of v. Thus, the length D line segment cannot be cut until the sidelength of the enclosing rectangle is at most $(15/2)D$. □

We proceed to a case analysis based on which of the two edges (v, l) or (v, f) is separated first by the decomposition. Observe that (v, l) can be separated for the first time by either a vertical or a horizontal line. If v and f are separated for the first time by a line produced by Sub-Rectangle, we assign v to f in E and the increase in cost is $5D/m$ by Lemma 3.3. Therefore we can assume for the remainder of the proof that v and f are first separated by a vertical cutting line produced by Cut-Rectangle. Let μ, ξ denote the lines separating for the first time (v, l) and (v, f) respectively.
CASE A. *Edge (v, l) is separated before (v, f).*

We will now calculate the expectation of the cost increase for the two possible subcases.
CASE A1. Edge (v, l) is separated for the first time by a vertical cutting line. We assign v to f in E. With some probability p, ξ has sidelength $L/2$ or more. By Lemma 3.4, ξ has sidelength at most $8D$ with probability $(1-p)$. Therefore, by Lemma 3.1, the cost increase is $O(D/m)$ with probability $(1-p)$. Now we turn to the case in which ξ has length $L/2$ or more. If μ is produced by Sub-Rectangle, by Lemma 3.3, μ has sidelength at most $5L$. If μ is produced by Cut-Rectangle, by Lemma 3.2 either ξ or μ has length at most $6L$. Moreover ξ has always length smaller than μ since it is produced second in the dissection. By Lemma 3.1 the cost increase is $O(L/m)$ regardless of the operation producing μ. By Fact 3.2, probability p is $O(D/L)$. Therefore the expected cost increase for CASE A1 is at most

$$(1-p)O(D/m) + pO(L/m) = O(D/m) + O((D/L)(L/m)) = O(D/m).$$

Remark 3.2 *The probability calculation for Case A1 depended only on the choice of which vertical line first cuts (v, f). This will be useful in Section 4 when we restrict our choices for vertical cutting lines.*

CASE A2. Edge (v,l) is separated for the first time by a horizontal cutting line. We assign v to f in E. We compute first the expectation of the cost increase conditioned upon the sidelength X of line μ. See Figure 1b. By Fact 3.1, (v,f) is cut by a line of sidelength Y with probability at most $3D/Y$. Observe that this is true regardless of the value of X. Moreover $L/2 \le Y \le X$ or by Lemma 3.4, $Y \le 8D$. The upper bound of X holds since (v,f) is contained in the rectangle cut by μ. For some constant c, the conditional expected cost increase is bounded by

$$\sum_{L/2 \le Y \le X | \exists i, Y=2^i} (cD/Y)(Y/m) = O\left((D/m)\log(X/L)\right).$$

We now remove the conditioning on X. If line μ was produced by Sub-Rectangle, by Lemma 3.3 it has length at most $5L$. No matter how large the probability of this event is, by the sum above the cost increase is $O(D/m)$. If line μ is produced by Cut-Rectangle, by Fact 3.1 it has sidelength X with probability at most $3L/X$. The expectation of the cost increase is at most

$$\sum_{X \ge L | \exists i, X=2^i} 3(L/X)\log(X/L)O(D/m) = O(D/m).$$

Remark 3.3 *To compute the probability of the event B that ξ has a given sidelength Y, we only used the total probability theorem. We partitioned the event space into the possible events $A_1, A_2, \ldots$ based on the different possible values of X and then applied $Pr[B] = \sum_i Pr[B|A_i]Pr[A_i]$. This remark will be useful in Section 4.*

CASE B. *Edge (v,f) is separated before (v,l).*
CASE B1. Edge (v,l) is separated for the first time by a vertical cutting line. The difference from CASE A1 is that we cannot argue that ξ has smaller sidelength than μ therefore we follow a different strategy. If the sidelength of ξ is at most mL we assign v to f else we assign v to l in E.

By Lemma 3.4, if the cost increase exceeds $8D/m$ the sidelength of ξ is $L/2$ or higher. By Fact 3.1, there is a constant c such that assigning v to f yields an expected cost increase of

$$c[(2D/L)(L/2m) + (D/L)(L/m) + (D/2L)(2L/m) + \ldots + (D/mL)(mL/m)] = O(D\log(m)/m).$$

By Fact 3.2, the probability that ξ has a sidelength of mL or more and hence that we assign v to l is $O(\frac{D}{mL})$. In this case Lemma 3.2 gives that the sidelength of μ is at most $3(L+D)$. Therefore, when v is assigned to l, the expected assignment cost (and not just the increase) is

$$O\left(\frac{D}{mL}(L + L/m)\right) = O(D/m + D/m^2).$$

Therefore, regardless of whether v is assigned to f or l the expected cost increase is $O(D\log(m)/m)$.
CASE B2. Edge (v,l) is separated for the first time by a horizontal cutting line. We follow the same strategy as in Case B1: if the sidelength of ξ is at most mL we assign v to f else we assign v to l in E.

By the same argument as in Case B1, if v is assigned to f the expected cost increase is $O(D\log(m)/m)$. Consider the case where v is assigned to l. If μ is produced by Sub-Rectangle by Lemma 3.3 it has sidelength at most $5L$ and hence the expected assignment cost is $O(D/m+D/m^2)$ by a calculation similar to Case B1.

If μ is produced by Cut-Rectangle it cannot have sidelength less than L. Moreover by Fact 3.1 it has sidelength X with probability at most $3L/X$. X cannot exceed the sidelength Y of ξ but for every possible X in the range $[L, Y]$, the conditional probability that μ has sidelength X is still at most $3L/X$. Therefore we obtain that the expected assignment cost for v is

$$O\left(\sum_{mL \le Y | \exists i, Y=2^i} \frac{D}{mL} \sum_{L \le X \le Y | \exists i, X=2^i} (L/X)(X + X/m)\right) = O\left(\frac{D}{mL}(L + L/m)\right) = O(D/m + D/m^2).$$

Remark 3.4 *To compute the probability of the event B that μ has a given sidelength X, we only used the total probability theorem. We partitioned the event space into the possible events $A_1, A_2, \ldots$ based on the different possible values of Y and then applied $Pr[B] = \sum_i Pr[B|A_i]Pr[A_i]$. This remark will be useful in Section 4.*

End of the proof of the Structure Theorem □

4 Modifying the Structure Theorem

The Structure Theorem in the previous section demonstrates that a portal-respecting $(1+O(\log(m)/m))$-approximate solution exists while only placing m portals on the boundary of the decomposition rectangles. Using ideas from [2], Theorem 3.1 would suffice by itself to design a dynamic programming algorithm running in, say, $O(2^{O(m)}kn^4)$ time. In this section we show how to effectively bound the number of rectangles to be enumerated by the dynamic program and thus obtain a nearly linear-time algorithm.

We give first some definitions. Consider the square of sidelength T that surrounds the original input. We know from Section 2 that $T = O(n^c)$, for some constant $c > 0$. We assume that $T = m2^\rho$ for some integer ρ and that the leftmost lower corner of the square lies at the origin of the axes. Call the vertical (horizontal) lines whose x-coordinate (y-coordinate) is an integral multiple of m *eligible.* Then, we call the vertical (horizontal) eligible lines with x-coordinate (y-coordinate) congruent to $0 \bmod 2^i$, $1 \le i \le \rho$, *i-allowable.* Note that the top and leftmost eligible lines are ρ-allowable, and that any j-allowable line is i-allowable for all $i < j$. A rectangle R of sidelength s is *t-allowable* if t is the maximum value such that: any two parallel sides of R of length s lie on t-allowable lines and $s \ge 2^t m$. Observe that when a rectangle is t-allowable, the aspect ratio is not bounded. The definition only guarantees that the smallest side has length $2^t m$. A rectangle which is t-allowable for some t and all whose sides have length within a factor of $5 + 1/2m$ of each other is called *allowable.*

We modify the Sub-Rectangle and Cut-Rectangle processes as follows.

Sub-Rectangle-new:
Input: An allowable rectangle containing at least one facility.
Process: Perform the Sub-rectangle process of the previous section. Let B_s be the computed rectangle.
Output: the minimal allowable rectangle that contains B_s.

Cut-Rectangle-new:
Input: An allowable rectangle containing at least one facility.
Process: Choose a cutting line in the middle third of the maximal side of the rectangle, uniformly at random among all lines that produce two allowable sub-rectangles.
Output: The two allowable sub-rectangles.

To illuminate further Cut-Rectangle-new consider an example where it takes as input a t-allowable rectangle with sidelength $m2^t$. This implies that the rectangle is a square. Let us consider the horizontal side as maximal. The middle third of the maximal side has $\lfloor m/3 \rfloor$ candidate t-allowable cutting lines. Choosing one, yields two sub-rectangles for which the horizontal side has length at least $(m/3)2^t$ and at most $(2/3)m2^t$. Accordingly the two subrectangles are each i-allowable for $i \geq t-2$.

Before proving the Modified Structure Theorem we examine the validity of the deterministic lemmata from Section 3 in the new setting. Clearly Lemma 3.2 continues to hold. We have to be more careful with Lemma 3.3, 3.4 due to the extra requirement that the output of Sub-Rectangle-new should be allowable. This might cause the rectangle output by Sub-Rectangle to be "stretched". We show first that this stretch is small.

Lemma 4.1 *Each side of the rectangle output by Sub-Rectangle-new has length at most* $1+2/m$ *the length of the corresponding side of the rectangle* B_s *computed during the process.*

Proof: Within the Sub-Rectangle process, the rectangle B' of sidelength s is grown by $s/3$ in each dimension. It follows that the lengths of the sides of B_s are within a factor of $(s+2s/3)/(2s/3)=5$ of each other. To meet the definition of an allowable rectangle, we only have to move the boundaries of B_s so that they fall on t-allowable lines. Here t is the maximum integer so that each side of B_s has length at least $2^t m$. To achieve this, we have to extend each side by at most 2^t in each direction for a total increase by a $1+2/m$ factor. □

Lemma 3.4 continues to hold with the modified decomposition. In fact it becomes slightly stronger since (cf. Proof of Lemma 3.4) the left boundary of any allowable rectangle produced by Sub-Rectangle-new that has sidelength $s < L/2$ and contains v is within distance at most $s(1/5+1/m)$ from v. We now give the equivalent to Lemma 3.3 in the modified setting.

Lemma 4.2 *Let* f *be a facility. If a cutting line segment* σ *produced by Sub-Rectangle-New intersects an assignment edge* (v,f) *of length* D, σ *has sidelength at most* $(5+6/m)D$.

Proof: We outline only the changes from the Proof of Lemma 3.3. Keeping the same notation, by Lemma 4.1 $(5/3)y \leq s \leq (5/3+2/m)y$. Since $y/3 < D$, we obtain $s < (5+6/m)D$. □

The difference between Lemma 3.3 and 4.2 is negligible from the point of view of the Modified Structure Theorem since it only introduces an additive $O(D/m^2)$ error term. We can now focus on the probabilistic part of the modified decomposition.

The new dissection is similar to the one from Section 3. The primary difference is that the randomization in the Cut-Rectangle process has been diminished. If we add back some randomization up front by shifting the original rectangle containing the input, we can get the same result as in the Structure Theorem on the expected increased cost of a portal respecting solution. We define an (a,b)-*shifted* coordinate system, $0 \leq a,b \leq T$ one in which the x and y coordinates are shifted by a and b respectively. The shifting uses wraparound as in [1]: a vertical line which had x-coordinate x_1 before the shifting has coordinate $x_1 + a \bmod T$ after. If a and b are chosen at random, any vertical or horizontal line is equally likely to be i-allowable. The *new modified cost* of an assignment is defined with respect to the new dissection give above.

Theorem 4.1 *(***Modified Structure Theorem***) Let* $m>0$ *be any integer and* $F \subset N$ *a set of open facilities. If the coordinate system is randomly shifted by* (a,b) *where* a *and* b *are chosen independently in* $[0,T]$ *then there is an assignment* E *such that the expected difference between the new modified cost of* E *and the greedy cost* C *under* F *is* $O(C\max\{1,\log(m)\}/m)$.

Proof: As mentioned, minor variations of the deterministic lemmata from Section 3 continue to hold in the restricted version of the dissection. The randomized portion in the proof of Theorem 3.1 only reasons about two types of events. Moreover, it reasons about each event in isolation (cf. Cases A1 and A2 in the proof, the rest are similar). Thus, we need only be concerned with the probabilities of each event. The random shift up front of the coordinate system along with the randomization inside the process will ensure that these two types of events occur in our decomposition into allowable rectangles with approximately the same probability as in the previous decomposition.

The first event (cf. Case A1) is that a line produced by Cut-Rectangle-new of length X cuts a line segment of length D. The probability of this event is required to be at most $3D/X$ in the proof of the Structure Theorem. We show that this continues to hold albeit with a constant larger than 3.

We assume for simplicity, that the line segment is in a rectangle R of sidelength exactly $X = m2^i$ at some point. (This will be true to within a constant factor.) If $D < 2^i$, we know that the segment is cut by at most one i-allowable line. The probability of this event is at most $D/2^i$ due to the random shift. The Cut-Rectangle-new process chooses from $m/3$ i-allowable lines uniformly at random. Thus, the line segment is cut with probability $1/(m/3)$ times $D/2^i$, which is $3D/X$ as required. If $D > 2^i$, we notice that D intersects at most $\lceil D/2^i \rceil < 2D/2^i$ i-allowable lines. By the union bound the probability that this line segment is cut during Cut-Rectangle-new on R is upperbounded by $2D/2^i$ times $3/m$, i.e., $6D/X$.

The second event (cf. Case A2) is the intersection of two events; a horizontal line of length X cuts a segment of length L and a vertical line of length Y cuts a segment of length D. In the proof of the Structure Theorem, the probability was shown by the total probability theorem to be upper bounded by the product of the probability bounds of the two events, i.e., $3L/X$ times $3D/Y$. The second term represented the conditional probability that edge (v, f) is cut by a line of sidelength Y given that (v, l) was cut by a horizontal line of sidelength X. We argued that the conditional probability does not depend on X.

For our restricted decomposition, the probability of each event in isolation can be bounded by $6L/X$ and $6D/Y$ as argued above. Moreover, we chose the horizontal and vertical shifts independently and we choose the horizontal and vertical cut-lines in different processes. Thus, we can also argue that the probability of the intersection of the two events is at most $6L/X$ times $6D/Y$. $\square$

We now prove a lemma bounding the number of allowable rectangles.

Lemma 4.3 *The number of allowable rectangles that contain l or more points from the input set N is $O(m^4(n/l)\log n)$.*

Proof: Our proof uses a charging argument. Let R_l be a rectangle on the plane that has minimum sidelength, say L and contains l points. We bound the cardinality of the set S_l of allowable rectangles, which are distinct from R_l, contain at least l points and have at least one point in common with R_l. Let R_a be such a rectangle. Then R_a has sidelength at least L, otherwise it would have been chosen instead of R_l.

We bound the number of allowable rectangles in S_l with sidelength $X \in [2^{i-1}, 2^i]m$ by $O(m^4)$ as follows. The corners must fall on the intersection of two t-allowable lines, that are within distance X from some side of R_l. Since allowable rectangles have bounded aspect ratio, the possible values for t are 2^{i-k}, $k = 0, 1, 2, 3$.

The number of j-allowable lines that are within distance X from some side of R_l is $O(X/2^{j-1})$ since $X \geq L$. Thus, the number of corner choices is $O(m^2)$. Two corners must be chosen, so the number of rectangles in S_l of sidelength $X \in [2^{i-1}, 2^i]m$ is $O(m^4)$. Since there are $O(\log n)$ values of i, $|S_l| = O(m^4 \log n)$.

Now, we remove R_l and its points from the decomposition, and repeat the argument on the remaining $n-l$ points. The number of repetitions until no points are left is $O(n/l)$ therefore by induction, we get a bound of $O(m^4(n/l)\log n)$ on the number of allowable rectangles that contain at least l points. □

5 The dynamic program

We have structural theorems relative to a particular decomposition. Unfortunately, the decompositions are defined with respect to the facility locations. However, in reality they only use the facility locations in the Sub-rectangle steps. Moreover, the number of sub-rectangles is at most polynomial in the size of the original rectangle. Indeed, the number of allowable sub-rectangles is polynomial in m and n. Thus, we can perform dynamic programming to find the optimal solution. The structure of the lookup table is similar to the one used in [2]. We exploit our Structure Theorem and the analysis on the total number of allowable rectangles to obtain a smaller number of entries.

The table will consist of a set of entries for each allowable rectangle that contains at least one point. For each allowable rectangle, we will also enumerate the following

- the number of facilities in the rectangle,
- a distance for each portal to the closest facility in the rectangle and
- a distance for each portal to the closest facility outside the rectangle.

Actually, we will only approximate the distances to the nearest facility inside and out of the rectangle to a precision of s/m for a rectangle of sidelength s. Moreover, we do not consider distances of more than $10s$. Finally, the distance value at a portal only changes by a constant from the distance value at an adjacent portal. This, will allow us to bound the total number of table entries corresponding to each allowable rectangle by $k2^{O(m)}$. See [2] for further details on the table construction.

We can compute the entries for a rectangle of sidelength s, by looking at either all the sub-rectangles from the Sub-rectangle process or by looking at all ways of cutting the rectangle into allowable smaller rectangles. This is bounded by $O(2^{O(m)})$ time per table entry. We bound the table size by noting that the total number of allowable rectangles is, by Lemma 4.3, at most $O(m^4 n\log n)$. Thus, we can bound the total number of entries in the table and the total computation time by $O(k2^{O(m)}n\log n)$.

We can improve this to $O(2^{O(m)}n\log k\log n)$ by a more careful estimation. By Lemma 4.3, the number of allowable rectangles that contain k or more points is $O(m^4(n/k)\log n)$. If an allowable rectangle contains fewer than $l<k$ points from N, we only need to keep $l2^{O(m)}$ entries for it, since at most l facilities can be placed inside it. Moreover, the number of allowable rectangles containing between l and $2l$ points is shown by Lemma 4.3 to be $O(m^4(n/l)\log n)$. We can now bound the total number of table entries by

$$k2^{O(m)}O(m^4(n/k)\log n)+\sum_{l=2^i}^{l<k}2^{O(m)}O(m^4(n/l)\log n)l=O(2^{O(m)}n\log k\log n).$$

We are now ready to state the main result of the paper.

Theorem 5.1 *Given an instance of the k-median problem in the 2-dimensional Euclidean space, and any fixed $m>0$, there is a randomized algorithm that computes a $(1+O(\log(m)/m))$-approximation, in expectation, with worst case running time $O(2^{O(m)}n\log k\log n)$.*

Repeating the algorithm $O(\log n)$ times gives a $(1 + O(\log(m)/m))$ approximation guarantee with probability $1 - o(1)$. The algorithm can be easily extended to instances in the d-dimensional Euclidean space. The main difference is that now we work with rectangular parallelepipeds whose faces lie on i-allowable hyperplanes. As a consequence the bound given in Lemma 4.3 increases by an $O(m^{2d-4})$ factor and in the dynamic program we have to maintain data for each of the $2d$ faces of each parallelepiped where each face contains m^{d-1} portals. Following the same steps as in the proof of Theorem 5.1 we obtain the following.

Theorem 5.2 *Given an instance of the k-median problem in the d-dimensional Euclidean space, and any fixed $m > 0$, there is a randomized algorithm that computes a $(1 + O(\log(m)/m))$-approximation, in expectation, with worst-case running time $O(2^{O(m^{d-1})} n \log k \log n)$.*

For the uncapacitated facility location problem we do not need to enumerate the number of facilities open for subrentagles. Only one number per rectangle, the minimum cost to open facilities, is enough. Hence we obtain immediately an approximation scheme with running time $O(2^{O(m^{d-1})} n \log n)$ if the input points satisfy the assumptions outlined in Section 2.

Theorem 5.3 *Given an instance of uncapacitated facility location in the d-dimensional Euclidean space, a polynomial in n approximation on the value of the service cost, and any fixed integer $m > 0$, there is a randomized algorithm that computes a $(1 + O(\log(m)/m))$-approximation, in expectation, with worst case running time $O(2^{O(m^{d-1})} n \log n)$.*

For any given accuracy $\varepsilon > 0$, we want $\ln(m)/m \leq c\varepsilon$, where m is the number of portals and c an appropriately small constant. Let us assume $c = 1$, since we can always decrease the given value of ε. Using the computer algebra package Maple [5] we obtain that for $m = \ln(1/\varepsilon)/(0.5\varepsilon)$, $\ln(m)/m \leq \varepsilon$ for any $\varepsilon \in (0, 0.5)$ while at the same time $m > 2$. On the other hand, setting $m = 3$, always yields an error less than 0.5. Therefore the asymptotic number of portals required for an accuracy of ε in the objective is

$$O(1 + \log(1/\varepsilon)/\varepsilon).$$

We obtain the following reinterpretations of Theorems 5.2, 5.3.

Theorem 5.4 *Given an instance of the k-median problem in the d-dimensional Euclidean space, and any fixed $\varepsilon > 0$, there is a randomized algorithm that computes a $(1 + \varepsilon)$-approximation, in expectation, with worst-case running time*

$$O\left(2^{O\left((1+\frac{\log(1/\varepsilon)}{\varepsilon})^{d-1}\right)} n \log k \log n\right).$$

Theorem 5.5 *Given an instance of the uncapacitated facility location problem in the d-dimensional Euclidean space, a polynomial in n approximation on the value of the service cost, and any fixed $\varepsilon > 0$, there is a randomized algorithm that computes a $(1 + \varepsilon)$-approximation, in expectation, with worst-case running time*

$$O\left(2^{O\left((1+\frac{\log(1/\varepsilon)}{\varepsilon})^{d-1}\right)} n \log n\right).$$

References

[1] S. Arora. Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. *Journal of the ACM*, 45:753–782, 1998.

[2] S. Arora, P. Raghavan, and S. Rao. Polynomial time approximation schemes for the euclidean k-medians problem. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 106–113, 1998.

[3] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[4] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[5] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt. *Maple V Language Reference Manual.* Springer-Verlag, 1991.

[6] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group Steiner tree and k-median. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 114–123, 1998.

[7] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.

[8] M. Charikar, S. Guha, E. Tardos, and D. Shmoys. A constant factor approximation algorithm for the k-median problem. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 1–10, 1999.

[9] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors, *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization*, volume 1412 of *LNCS*, pages 180–194. Springer-Verlag, Berlin, 1998.

[10] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. Mirchandani and R. Francis, editors, *Discrete Location Theory.* John Wiley and Sons, Inc., New York, 1990.

[11] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 434–444, 1988.

[12] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[13] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 649–657, 1998.

[14] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22:148–162, 1982.

[15] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48:274 – 296, 2001.

[16] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-median problem. In J. Nešetřil, editor, *Proceedings of the 7th Annual European Symposium on Algorithms*, volume 1643 of *Lecture Notes in Computer Science*, pages 378–389. Springer-Verlag, 1999.

[17] J. H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44:245–249, 1992.

[18] J. H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 771–782, 1992.

[19] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28:1298–1309, 1999.

[20] S. Rao and W. D. Smith. Improved approximation schemes for geometrical graphs via *spanners* and *banyans*. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 540–550, 1998.

[21] D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 265–274, 1997.