On the Fixed-Parameter Tractability of Capacitated Clustering

Vincent Cohen-Addad
CNRS & Sorbonne Université, Paris, France

Jason Li
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We study the complexity of the classic capacitated k-median and k-means problems parameterized by the number of centers, k. These problems are notoriously difficult since the best known approximation bound for high dimensional Euclidean space and general metric space is $\Theta(\log k)$ and it remains a major open problem whether a constant factor exists.

We show that there exists a $(3+\epsilon)$-approximation algorithm for the capacitated k-median and a $(9+\epsilon)$-approximation algorithm for the capacitated k-means problem in general metric spaces whose running times are $f(\epsilon, k)n^{O(1)}$. For Euclidean inputs of arbitrary dimension, we give a $(1+\epsilon)$-approximation algorithm for both problems with a similar running time. This is a significant improvement over the $(7+\epsilon)$-approximation of Adamczyk et al. for k-median in general metric spaces and the $(69+\epsilon)$-approximation of Xu et al. for Euclidean k-means.

2012 ACM Subject Classification Theory of computation → Facility location and clustering; Theory of computation → Fixed parameter tractability; Mathematics of computing → Probabilistic algorithms; Mathematics of computing → Dimensionality reduction

Keywords and phrases approximation algorithms, fixed-parameter tractability, capacitated, k-median, k-means, clustering, core-sets, Euclidean

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.41

Category Track A: Algorithms, Complexity and Games

Funding *Vincent Cohen-Addad*: Ce projet a bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du Programme Appel à projets générique JCJC 2018 portant la référence suivante: ANR-18-CE40-0004-01.
Jason Li: Supported in part by NSF awards CCF-1536002, CCF-1540541, and CCF-1617790.

1 Introduction

Clustering under capacity constraints is a fundamental problem whose complexity is still poorly understood. The capacitated k-median and k-means problems have attracted a lot of attention over the recent years (*e.g.*: [4, 22, 23, 24, 13, 3, 8, 6]), but the best known approximation algorithm for capacitated k-median remains a somewhat folklore $O(\log k)$-approximation using the classic technique of embeddings the metric space into trees that follows from the work of Charikar et al [5] on the uncapacitated version, see also [1] for a complete exposition.

Arguably, the hardness of the problem comes from having both a hard constraint on the number of clusters, k, and on the number of clients that can be assigned to each cluster. Indeed, constant factor approximation algorithms are known if the capacities [22, 23] or the number of clusters can be violated by a $(1+\epsilon)$ factor [4, 13], for constant ϵ. Moreover, the capacitated facility location problem admits constant factor approximation algorithms with no capacity violation. On the other hand and perhaps surprisingly, the best known lower bound for capacitated k-median is not higher than the $1+2/e$ lower bound for the uncapacitated version of the problem.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 41; pp. 41:1–41:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Thus, to improve the understanding of the problem a natural direction consists in obtaining better approximation algorithms in some specific metric spaces, or through the fixed-parameter complexity of the problem. For example, a quasi-polynomial time approximation scheme (QPTAS) for capacitated k-median in Euclidean space of fixed dimension with $(1+\epsilon)$ capacity violation was known since the late 90's [2]. This has been recently improved to a PTAS for $\mathbb{R}^2$ and a QPTAS for doubling metrics without capacity violation [9]. It remains an interesting open question to obtain constant factor approximation for other metrics such as planar graphs or Euclidean space of arbitrary dimension.

For many optimization problems are at least W[1]-hard and so obtaining exact fixed-parameter tractable (FPT) algorithms is unlikely. However, FPT algorithms have recently shown that they can help break long-standing barriers in the world of approximation algorithms. FPT approximation algorithms achieving better approximation guarantees than the best known polynomial-time approximation algorithms for some classic W[1]- and W[2]-hard problems have been designed. For example, for k-cut [15], for k-vertex separator [21] or k-treewidth-deletion [16].

For the fixed-parameter tractability of the k-median and k-means problems, a natural parameter is the number of clusters k. The FPT complexity of the classic uncapacitated k-median problem, parameterized by k, has received a lot of attention over the last 15 years. From a lower bound perspective, the problem is known to be W[2]-hard in general metric spaces and assuming the exponential time hypothesis (ETH), even for points in $\mathbb{R}^4$, there is no exact algorithm running in time $n^{o(k)}$ [10]. For $\mathbb{R}^2$ there exists an exact $n^{O(\sqrt{k})}$ which is the best one can hope for assuming ETH [10], see also [26].

From an upper bound perspective, *coreset* constructions and PTAS with running time $f(k,\epsilon)n^{O(1)}$ have been known since the early 00's [12, 19, 17, 18, 14]. In the language of fixed-parameter tractability, a coreset is essentially an "approximate kernel" for the problem: given a set P of n points in a metric space, a coreset is, loosely speaking, a mapping from the points in P to a set of points Q of size $(k \log n\epsilon^{-1})^{O(1)}$ such that any clustering of Q of cost γ can be converted into a clustering of P of cost at most $\gamma \pm \epsilon \mathsf{cost}(\mathrm{OPT})$, through the inverse of the mapping (where OPT is the optimal solution for P). See Definition 9 for a more complete definition.

In Euclidean space, several coreset constructions for uncapacitated k-median are independent of the input size and of the dimension and so are truly approximate kernels. Thus approximation schemes can simply be obtained by enumerating all possible partitions of the coreset points into k parts, evaluating the cost of each of them and outputing the one of minimum cost. However, obtaining similar results in general metric spaces seems much harder and is likely impossible. In fact, obtaining an FPT approximation algorithm with approximation guarantee less than $1 + 2/e$ is impossible assuming Gap-ETH, see [11].

For the capacitated k-median and k-means problems much less is known. First, the coreset constructions or the classic FPT-approximation schemes techniques of [20, 12] do not immediately apply. Thus, very little was known until the recent result of Adamczyk et al. [1] who proposed a $(7+\epsilon)$-approximation algorithm running in time $k^{O(k)}n^{O(1)}$. More recently, a $(69+\epsilon)$-approximation algorithm for the capacitated k-means problem with similar running time has been proposed by Xu et al. [28].

1.1 Our Results

We present a coreset construction for the capacitated k-median and k-means problems, with general capacities, and in general metric spaces (Theorem 11). For an n points set, the coreset has size $\mathrm{poly}(k\epsilon^{-1} \log n)$.

From this we derive a $(3+\epsilon)$-approximation for the k-median problem and a $(9+\epsilon)$-approximation for the k-means problem in general metric spaces.

▶ **Theorem 1.** *For any $\epsilon > 0$, there exists a $(3+\epsilon)$-approximation algorithm for the capacitated k-median problem and a $(9+\epsilon)$-approximation algorithm for the capacitated k-means problem running in time $(k\epsilon^{-1}\log n)^{O(k)}n^{O(1)}$. This running time can also be bounded by $(k/\epsilon)^{O(k)}n^{O(1)}$.*

This results in a significant improvement over the recent results of Adamczyk et al. [1] for k-median and Xu et al. [28] for (Euclidean) k-means, in the same asymptotic running time.

Moreover, combining with the techniques of Kumar et al. [20], we obtain a $(1+\epsilon)$-approximation algorithm for points in $\mathbb{R}^d$, where d is arbitrary. We believe that this is an interesting result: while it seems unlikely that one can obtain an FPT-approximation better than $1+2/e$ in general metrics, it is possible to obtain an FPT-$(1+\epsilon)$-approximation in Euclidean metrics of arbitrary dimension. This works for both the *discrete* and *continuous* settings: in the former, the set of centers must be chosen from a discrete set of candidate centers in $\mathbb{R}^d$ and the capacities may not be uniform, while in the latter the centers can be placed anywhere in $\mathbb{R}^d$ and the capacities are uniform.

▶ **Theorem 2.** *For any $\epsilon > 0$, there exists a $(1+\epsilon)$-approximation algorithm for the discrete, Euclidean, capacitated k-means and k-median problems which runs in time $(k\epsilon^{-1}\log n)^{k\epsilon^{-O(1)}}$ $n^{O(1)}$. This running time can also be bounded by $(k\epsilon^{-1})^{k\epsilon^{-O(1)}}$ $n^{O(1)}$.*

▶ **Theorem 3.** *For any $\epsilon > 0$, there exists a $(1+\epsilon)$-approximation algorithm for the continuous, Euclidean, capacitated k-means and k-median problems running in time $(k\epsilon^{-1}\log n)^{k\epsilon^{-O(1)}}$ $n^{O(1)}$. This running time can also be bounded by $(k\epsilon^{-1})^{k\epsilon^{-O(1)}}n^{O(1)}$.*

These two results are a major improvement over the 69-approximation algorithm of Xu et al. [28].

1.2 Preliminaries

We now provide a more formal definition of the problems.

▶ **Definition 4.** *Given a set of points V in a metric space with distance function d, together with a set of* clients *$C \subseteq V$, a set of* centers *$\mathbb{F} \subseteq V$ with a* capacity *$\eta_f \in \mathbb{Z}_+$ for each $f \in \mathbb{F}$, and an integer k, the* capacitated k-median problem *asks for a set $F \subseteq \mathbb{F}$ of k* centers *and an assignment $\mu : C \mapsto F$ such that $\forall f \in F$, $|\{c \mid \mu(c) = f\}| \le \eta_f$ and that minimizes $\sum_{c\in C} d(c, \mu(c))$. We abbreviate the capacitated k-median instance as $((V,d), C, \mathbb{F}, k)$.*

▶ **Definition 5.** *The* capacitated k-means problem *is identical, except we seek to minimize $\sum_{c\in C} d(c, \mu(c))^2$.*

In the literature, centers are sometimes called *facilities*, but we will use *centers* throughout for consistency.

In the case of the capacitated Euclidean k-median and k-means, our approach works for the two main definitions. First, the definition of [28, 20]: $P = \mathbb{R}^d$ and capacities are uniform, namely $\eta_f = \eta_{f'}$, $\forall f, f' \in \mathbb{R}^d$. Second, P is some specific set of points in $\mathbb{R}^d$, and for each $f \in P$, the input specifies a specific capacity η_f

▶ **Definition 6.** *Given a capacitated k-median instance $((V,d), C, \mathbb{F}, k)$ and a set of chosen centers $F \subseteq \mathbb{F}$, define $\mathsf{CapKMed}(C, F)$ as the cost of the optimal assignment of the clients to the chosen centers. If it is impossible, i.e., the sum of the capacities of the centers is less than $|C|$, then $\mathsf{CapKMed}(C, F) = \infty$.*

In our analysis, we will also encounter formulations where the clients have positive *real* weights. In this case, we define a *fractional* variant of capacitated k-median, where the assignment μ is allowed to be fractional.

▶ **Definition 7.** *Suppose the clients also have weights, so we are given clients C and a weight function $w : C \to \mathbb{R}_+$. Let $W \subseteq C \times \mathbb{R}_+$ be the set of pairs $\{(c, w(c)) : c \in C\}$. Then,* $\mathsf{FracCapKMed}(W, F)$ *is the minimum value of $\sum_{c\in C, f\in F} \mu(c, f)\, d(c, f)$ over all "fractional assignments" $\mu : C \times F \to \mathbb{R}_+$ such that:*

1. *$\forall c \in C$, $\sum_{f\in F} \mu(c, f) = w(c)$, i.e., μ is a proper assignment of clients, and*
2. *$\forall f \in F$, $\sum_{c\in C} \mu(c, f) \le \eta_f$, i.e., μ satisfies capacity constraints at all centers.*

▶ **Definition 8.** *We define* $\mathsf{CapKMeans}(C, F)$ *and* $\mathsf{FracCapKMeans}(W, F)$ *similarly, except our objective functions are $\sum_{c\in C} d(c, \mu(c))^2$ and $\sum_{c\in C, f\in F} \mu(c, f)\, d(c, f)^2$, respectively.*

It is well-known that, given a set $F \subseteq \mathbb{F}$ of centers, the problem of finding the optimum μ is an (integral) *minimum-cost flow* problem, which can be solved in polynomial time. Therefore, we assume that every time we have a set $F \subseteq \mathbb{F}$, we can evaluate $\mathsf{CapKMed}(C, F)$ and $\mathsf{CapKMeans}(C, F)$ in polynomial time. Similarly, FracCapKMed and FracCapKMeans can be solved through fractional min-cost flow, or even an LP, in polynomial time. Furthermore, if W is exactly the set C of clients with weight 1, i.e., $W = \{(c, 1) : c \in C\}$, then $\mathsf{CapKMed}(C, F) = \mathsf{FracCapKMed}(W, F)$, since the min-cost flow formulation of FracCapKMed has integral capacities and therefore integral flows as well.

We now formally state our definition of coresets, sometimes called *strong* coresets in the literature.

▶ **Definition 9.** *A (strong) coreset for a capacitated k-median instance $((V, d), C, \mathbb{F}, k)$ is a set of weighted clients $W \subseteq C \times \mathbb{R}_+$ such that for every set of centers $F \subseteq \mathbb{F}$ of size k,*

$$\mathsf{FracCapKMed}(W, F) \in (1 - \epsilon, 1 + \epsilon) \cdot \mathsf{CapKMed}(C, F).$$

The definition is identical for capacitated k-means, except CapKMed *and* FracCapKMed *are replaced by* CapKMeans *and* FracCapKMeans *above.*

▶ **Fact 10.** *Let W be a coreset for a capacitated k-median instance $((V, d), C, \mathbb{F}, k)$. We have*

$$\min_{\substack{F\subseteq\mathbb{F}\\|F|=k}} \mathsf{FracCapKMed}(W, F) \in (1 - \epsilon, 1 + \epsilon) \cdot \min_{\substack{F\subseteq\mathbb{F}\\|F|=k}} \mathsf{CapKMed}(C, F),$$

In particular, an α-approximation of $\min_{F\subseteq\mathbb{F}, |F|=k} \mathsf{FracCapKMed}(W, F)$ implies a $(1+O(\epsilon))\alpha$-approximation to the capacitated k-median instance. The same holds in the capacitated k-means case, with FracCapKMed *and* CapKMed *replaced by* FracCapKMeans *and* CapKMeans*, respectively.*

For a capacitated k-median or k-means instance $((V, d), C, \mathbb{F}, k)$, the *aspect ratio* is the ratio of the maximum and minimum distances between any two points in $C \cup F$. It is well-known that we may assume, with a multiplicative error of $(1 + o(1))$ in the optimal solution, that the instance has poly(n) aspect ratio.[1] Therefore, we will make this assumption throughout the paper.

[1] For example, the following modification to the distances d does the trick. First, compute an $O(\log k)$-approximation [5] to the problem, and let that value be M. For any two points $u, v \in C \cup F$ with $d(u, v) > Mn^{10}$, truncate their distance to exactly Mn^{10}. Then, add Mn^{-10} distance to each pair of points $u, v \in C \cup F$. The aspect ratio is now bounded by $O(n^{20})$.

Lastly, we define $\mathbb{R}_+$ and $\mathbb{Z}_+$ as the set of positive reals and positive integers, respectively. As usual, we define *with high probability (w.h.p.)* as with probability $1-n^{-Z}$ for an arbitrarily large positive constant Z, fixed beforehand.

2 Coreset for k-median

In this section, we prove our main technical result for the k-median case: constructing a coreset for capacitated k-median of size $\text{poly}(k \log n\, \epsilon^{-1})$.

▶ **Theorem 11.** *For any small enough constant $\epsilon \geq 0$, there exists a Monte Carlo algorithm that, given an instance $((V,d), C, \mathbb{F}, k)$ of capacitated k-median, outputs a (strong) coreset $W \subseteq C$ with size $O(k^2 \log^2 n/\epsilon^3)$ in polynomial time, w.h.p.*

▶ **Theorem 12.** *For any small enough constant $\epsilon \geq 0$, there exists a Monte Carlo algorithm that, given an instance $((V,d), C, \mathbb{F}, k)$ of capacitated k-means, outputs a (strong) coreset $W \subseteq C$ with size $O(k^5 \log^5 n/\epsilon^3)$ in polynomial time, w.h.p.*

Our inspiration for the coreset construction is Chen's algorithm [7] based on random sampling. Our algorithm is essentially the same, with slightly worse bounds in the sampling step, although our analysis is a lot more involved. We describe the full algorithm in pseudocode below (see Algorithm 1).

At a high level, the algorithm first partitions the client set C into $\text{poly}(k, \log n)$ many subsets, called *rings*, with the help of a polynomial-time approximate solution (see line 1). The sets are called rings because they are of the form $C_i \cap (\mathsf{ball}(f_i', R) \setminus \mathsf{ball}(f_i', R/2))$ for some subset of clients $C_i \subseteq C$, some facility $f_i' \in \mathbb{F}$, and some positive number R (see line 7). Then, for each ring $C_{i,R}$, if $|C_{i,R}|$ is small enough, the algorithm adds the entire ring into the coreset (each with weight 1); otherwise, the algorithm takes a random sample of $r = \text{poly}(k, \log n)$ many clients in $C_{i,R}$, weights each sampled client by $|C_{i,R}|/r$, and adds the weighted sample to the coreset. The weighting ensures that the total weight of the sampled points is always equal to $|C_{i,R}|$. To prove that the algorithm produces a coreset w.h.p., Chen union bounds over all $\binom{|\mathbb{F}|}{k}$ choices of a set of k facilities, and shows that for each choice $F \subseteq \mathbb{F}$, with probability at least $1 - n^{-\Omega(k)}$, the total cost to assign the coreset points to F is approximately the total cost to assign the original clients C to F; this statement is proved through standard concentration bounds. More details and intuition for the algorithm can be found in Section 3 of Chen's paper [7].

2.1 Single ring case

We first restrict ourselves to sampling from a *single* ring $C_{i,R} \subseteq C$. That is, while we still consider the cost of serving the clients outside of $C_{i,R}$, we only perform the sampling (lines 12–13) on one ring $C_{i,R}$. The general case of $O(k \log n)$ many rings is more complicated than simply treating each ring separately. Due to space constraints, we only consider the single ring case in this extended abstract, and the rest is deferred to the full version.

Fix an arbitrary ring $C_{i,R}$ throughout this section, and define $C' := C_{i,R}$ for convenience. Let $N := |C'|$ be the number of clients, and let $f' := f_i'$ be the ring center of C' (line 4). Let W' be the (weighted) centers in $C_{i,R}$ sampled by the algorithm (lines 12–13), together with the (unweighted) centers in $C \setminus C'$, which have weight 1. Our goal is to show that $\mathsf{FracCapKMed}(W', F)$, the cost after sampling only from C', is close to the original cost $\mathsf{CapKMed}(C, F)$.

Algorithm 1 CoreSet(I).

1: $F' = \{f'_1, \ldots, f'_{O(k)}\} \leftarrow$ an $(O(1), O(1))$ bicriteria solution to instance I, namely a capacitated $O(k)$-median solution with total cost $ALG' \leq O(OPT)$ ▷ using, e.g., [23]
2: $W \leftarrow \emptyset$ ▷ $W \subseteq C \times \mathbb{R}_+$ is the final coreset at the end of the algorithm
3: Define $d_{\min}$ and $d_{\max}$ as the minimum and maximum distances, respectively, between any two points in $C \cup \mathbb{F}$ ▷ $d_{\max}/d_{\min}$ is the aspect ratio
4: **for** each center f'_i **do** ▷ $O(k)$ centers
5: $C_i \leftarrow$ the clients in C assigned to center f'_i
6: **for** each R, a power of 2 in the range $[d_{\min}, 2d_{\max}]$ **do** ▷ $O(\log n)$ iterations, assuming poly(n) aspect ratio
7: $C_{i,R} \leftarrow C_i \cap (\mathsf{ball}(f'_i, R) \setminus \mathsf{ball}(f'_i, R/2))$ ▷ We call the sets $C_{i,R}$ *rings*, with *ring center* f'_i. The rings $C_{i,R}$ over all i, R partition the client set C.
8: $r \leftarrow \gamma k \log n/\epsilon^3$ for sufficiently large (absolute) constant γ
9: **if** $|C_{i,R}| \leq r$ **then**
10: add $(c, 1)$ to W for each $c \in C_{i,R}$ ▷ $C_{i,R}$ small enough: add everything into coreset
11: **else**
12: sample r random centers in $C_{i,R}$ (without replacement)
13: add $(c, \frac{|C_{i,R}|}{r})$ to W for each sampled center c ▷ weighted so that total weight is still $|C_{i,R}|$

▶ **Lemma 13.** *W.h.p., for any set of k centers $F \subseteq \mathbb{F}$ satisfying* $\mathsf{CapKMed}(C, F) < \infty$*,*

$$|\mathsf{FracCapKMed}(W', F) - \mathsf{CapKMed}(C, F)| \leq \epsilon N R. \quad (1)$$

It is clear that the output W has size $O(k^2 \log^2 n/\epsilon^3)$. The rest of this section focuses on proving that W is indeed a coreset, w.h.p.

The intuition behind the ϵNR additive error is that we can "charge" this error to the cost of the bicriteria solution (line 1) that C' is responsible for. In particular, the total cost of assigning clients in C' to ring center f' in the bicriteria solution is at least $N \cdot R/2$, since all clients in C' are distance at least $R/2$ to f'. Therefore, we charge an additive error of ϵNR to a $NR/2$ portion of ALG', which is a "rate" of 2ϵ to 1. If we can do the same for all rings, then since the portions of ALG' sum to ALG', our total additive error is at most $2\epsilon \cdot ALG' = O(\epsilon) \cdot OPT$. Finally, replacing ϵ with a small enough $\Theta(\epsilon)$ gives the desired additive error of $\epsilon \cdot OPT$; note that this is where we use that the approximation ratio of ALG' is $O(1)$, and that the specific approximation ratio is not important (as long as it is constant). The formalization of this intuition is deferred to the full version; the argument is identical to Chen's [7], so we claim no novelty here.

We now prove Lemma 13. First of all, if $N = |C'| \leq r$ (line 9), then sampling changes nothing, and $\mathsf{FracCapKMed}(W', F) = \mathsf{CapKMed}(C, F)$. Therefore, for the rest of the proof, we assume that $N > r = \gamma k \log n/\epsilon^3$, with the γ taken to be a large enough constant.

Our high-level strategy is the same as Chen's: we union bound over all sets of centers $F \subseteq \mathbb{F}$ of size k, and prove that for a fixed set F, the probability of violating (1) is at most $n^{-(k+10)}$.[2] Union bounding over all $\leq \binom{n}{k}$ choices of F gives probability $\leq n^{-10}$ of

[2] For simplicity of presentation, we will focus on a success probability of $1 - n^{-10}$. The constants can be easily tweaked so that the algorithm succeeds w.h.p., i.e., with probaility $1 - n^{-Z}$ for any positive constant Z.

violating (1), proving the lemma. Therefore, from now on, we focus on a single, arbitrary set $F \subseteq \mathbb{F}$ of size k satisfying $\mathsf{CapKMed}(C, F) < \infty$, and aim to show that (1) fails with probability $\leq n^{-(k+10)}$.

For our analysis, we define a function $g : \mathbb{R}_+^{C'} \to \mathbb{R}_+$ as follows. For an input vector $\mathbf{d} \in \mathbb{R}_+^{C'}$ (indexed by clients in C'), consider a min-cost flow instance $\mathsf{FlowInstance}(\mathbf{d})$ on the graph metric with the following demands: set demand d_c at each client $c \in C'$, demand 1 at each client $c \in C \setminus C'$, and demand $N - \sum_{c \in C'} d_c$ (this demand can be negative) at ring center $f' = f'_i$ (so we are effectively treating f' as a special client with possibly negative demand, not a facility). Observe that $\mathsf{FlowInstance}(\mathbf{d})$ is a feasible min-cost flow instance, because the sum of demands is exactly

$$\sum_{c \in C'} d_c + |C \setminus C'| + \left(N - \sum_{c \in C'} d_c\right) = |C \setminus C'| + N = |C|,$$

which is the same as the sum of demands in the instance $\mathsf{CapKMed}(C, F)$, which is feasible by assumption.

Given this setup for an input vector $\mathbf{d} \in \mathbb{R}_+^{C'}$, we define the function $g(\mathbf{d})$ as the min-cost flow of $\mathsf{FlowInstance}(\mathbf{d})$. Observe that $g(\mathbb{1})$ is exactly $\mathsf{CapKMed}(C, F)$.

Now define a random vector $X \in \mathbb{R}_+^{C'}$ as follows. Each coordinate of X is independently N/r with probability r/N and 0 otherwise, so that $\mathbb{E}[X] = \mathbb{1}$. Note that X does not accurately represent our sampling of r clients, since this process is not guaranteed to sample exactly r clients. Nevertheless, it is intuitively clear that with probability $\Omega(1/n)$, X will indeed have exactly r nonzero entries, since r is the expected number; we prove this formally in the following simple claim (with $p = r/N$), whose routine proof is deferred to the full version. And if we *condition* on this event, then $g(X)$ and $\mathsf{CapKMed}(C, F)$ are now identically distributed.

▷ Claim 14. Let N be a positive integer, and let $p \in (0, 1)$ such that pN is an integer. The probability that $\mathsf{Binomial}(N, p) = pN$ is at least $\Omega(1/\sqrt{N})$.

In light of all this, our main argument has two steps. First, we show that $g(X)$ is concentrated around $\mathbb{E}[g(X)]$ using martingales. However, what we really need is concentration around $g(\mathbb{E}[X]) = g(\mathbb{1}) = \mathsf{CapKMed}(C, F)$, so our second step is to show that $\mathbb{E}[g(X)] \approx g(\mathbb{E}[X])$ (with probability 1). We formally state the lemmas below which, as discussed, together imply Lemma 13.

▶ **Lemma 15.** *Assume that $|C'| > \Theta(k \log n/\epsilon^3)$. With probability $\geq 1 - n^{-(k+20)}$, we have $|g(X) - \mathbb{E}[g(X)]| \leq \epsilon NR/2$.*

▶ **Lemma 16.** *Assume that $|C'| > \Theta(k \log n/\epsilon^3)$. Then, $|\mathbb{E}[g(X)] - g(\mathbb{E}[X])| \leq \epsilon NR/2$.*

2.1.1 Proof of Lemma 15: concentration around $\mathbb{E}[g(X)]$ via martingales.

To show that $g(X)$ is concentrated around its mean, we show that g is sufficiently Lipschitz (w.r.t. the ℓ_1 distance in $\mathbb{R}_+^{C'}$), and then apply standard martingale tools.

▷ Claim 17. The function g is R-Lipschitz w.r.t. the ℓ_1 distance in $\mathbb{R}_+^{C'}$.

Proof. Fix a client $c \in C'$, and consider two vectors $\mathbf{d}, \mathbf{d}' \in \mathbb{R}_+^{C'}$ with $\mathbf{d}' = \mathbf{d} + \delta \cdot \mathbb{1}_c$. By definition of $\mathsf{FlowInstance}$, the only difference between $\mathsf{FlowInstance}(\mathbf{d})$ and $\mathsf{FlowInstance}(\mathbf{d}')$ is that in $\mathsf{FlowInstance}(\mathbf{d}')$, client c has δ more demand and "special client" f' has δ less demand. Therefore, if we begin with the min-cost flow of $\mathsf{FlowInstance}(\mathbf{d})$, and then add δ units of flow from c to f', then we now have a feasible flow for $\mathsf{FlowInstance}(\mathbf{d}')$.[3] This means that

$$g(\mathbf{d}') \le g(\mathbf{d}) + \delta R.$$

Similarly, starting from a min-cost flow of $\mathsf{FlowInstance}(\mathbf{d}')$ and then adding δ units of flow from f' to c, we obtain a feasible flow for $\mathsf{FlowInstance}(\mathbf{d})$, so

$$g(\mathbf{d}) \le g(\mathbf{d}') + \delta R.$$

Together, these two inequalities show that g is R-Lipschitz. ◁

We state the following Chernoff bound for Lipschitz functions, which can be proven by adapting the standard (multiplicative) Chernoff bound proof to a martingale.

► **Theorem 18.** *Let $x_1, \dots, x_n$ be independent random variables taking value b with probability p and value 0 with probability $1-p$, and let $g : [0,1]^n \to \mathbb{R}$ be a L-Lipschitz function in ℓ_1 norm. Define $X := (x_1, \dots, x_n)$ and $\mu := \mathbb{E}[g(X)]$. Then, for $0 \le \epsilon \le 1$:*

$$\Pr\left[\left|g(X) - \mathbb{E}[g(X)]\right| \ge \epsilon pnbL\right] \le 2e^{-\epsilon^2 pn/3}$$

We apply Theorem 18 on the L-Lipschitz function g with the randomly sampled demands. Set $p := r/N$ as the sampling probability, so that $X \in \{0, 1/p\}^N$ is the random demand vector. Setting $n := N$, $b := 1/p$, and $L := R$, we obtain

$$\begin{aligned}
&\Pr\left[\left|g(X) - \mathbb{E}[g(X)]\right| \ge (\epsilon/2)NR\right] \\
&\quad = \Pr\left[\left|g(X) - \mathbb{E}[g(X)]\right| \ge (\epsilon/2)pnbL\right] \\
&\quad \le 2\exp\left(\frac{-(\epsilon/2)^2 pn}{3}\right) \\
&\quad = 2\exp\left(\frac{-(\epsilon/2)^2 (r/N)N}{3}\right) = \exp\left(-\Theta(\epsilon^2 r)\right) = \exp\left(-\Omega(\epsilon^2 \cdot \frac{k\log n}{\epsilon^2})\right) \\
&\quad \le n^{-(k+20)}
\end{aligned}$$

for sufficiently large γ in the definition of $r = \gamma k \log n/\epsilon^2$. This concludes Lemma 15.

2.1.2 Proof of Lemma 16: relating $\mathbb{E}[g(X)]$ with $g(\mathbb{E}[X])$.

We have obtained concentration about $\mathbb{E}[g(X)]$, but we really need concentration around $g(\mathbb{E}[X]) = \mathsf{CapKMed}(C', F)$. We establish this by proving Lemma 16.

We first show the easy direction, that $g(\mathbb{E}[X]) \le \mathbb{E}[g(X)]$, which essentially follows from the convexity of min-cost flow: Suppose the outcomes of random variable X are $\mathbf{d}_1, \mathbf{d}_2, \dots$ with respective probabilities $\mu_1, \mu_2, \dots$, so that $\mathbb{E}[g(X)] = \sum_i \mu_i g(\mathbf{d}_i)$. Now consider the flow obtained by adding up, for each i, the min-cost flow of $\mathsf{FlowInstance}(\mathbf{d}_i)$ scaled by μ_i. This flow is a feasible flow to $\mathsf{FlowInstance}(\mathbb{E}[X])$ and has cost at most $\mathbb{E}[g(X)]$. Since the min-cost flow of $\mathsf{FlowInstance}(\mathbb{E}[X])$ can only be lower, we have $g(\mathbb{E}[X]) \le \mathbb{E}[g(X)]$.

We now prove the other direction: $\mathbb{E}[g(X)] \le g(\mathbb{E}[X]) + \epsilon NR/2$.

[3] We define demand so that if a vertex v has $d > 0$ demand, then d flow must exit v in a feasible flow, and if it has $d < 0$ demand, then $|d|$ flow must enter v.

▷ Claim 19. With probability 1, $g(X) \le g(\mathbb{E}[X]) + nNR$.

Proof. Since $X \in [0, N/r]^N$, and since g is R-Lipschitz, the entire range of $g(X)$ is contained in some interval of length $N \cdot N/r \cdot R \le N \cdot n \cdot R$. Since $\mathbb{E}[X] \in [0, N/r]^N$ as well, the value $g(\mathbb{E}[X])$ is also contained in that interval. The statement follows. ◁

▶ **Lemma 20.** *With probability* $\ge 1 - n^{-10}$, $g(X) \le g(\mathbb{E}[X]) + 0.49\epsilon NR$.

Due to space constraints, the proof of Lemma 20, which is long and technical, is deferred to the full version. Assuming Lemma 20, we now show how Claim 19 and Lemma 20 together imply Lemma 16: we have

$$\begin{aligned}\mathbb{E}[g(X)] &\le n^{-10} \cdot \big(g(\mathbb{E}[X]) + nNR\big) + (1 - n^{-10})\big(g(\mathbb{E}[X]) + 0.49\epsilon NR\big)\\ &= g(\mathbb{E}[X]) + \big(n^{-10} \cdot n + (1 - n^{-10}) \cdot 0.49\epsilon\big)NR\\ &\le g(\mathbb{E}[X]) + (\epsilon/2)NR,\end{aligned}$$

finishing the proof of Lemma 16.

2.2 $(3 + \epsilon)$- and $(9 + \epsilon)$-approximation – Proof of Theorem 1

In this section, we finish the algorithm for Theorem 1. We will focus mainly on the k-median case, since the k-means case is nearly identical.

Suppose we run the coreset for the capacitated k-median instance with parameter ϵ_0 (to be set later), obtaining a coreset $W \subseteq C \times \mathbb{R}^+$ of size $\mathrm{poly}(k \log n\, \epsilon_0^{-1})$. We now want to compute some $F \subseteq \mathbb{F}$ of size k and an assignment μ of the clients in W to F minimizing $\sum_{(c,w)\in W} w \cdot d(c, \mu(c))$. By definition of coreset, if we compute an α-approximation to this problem, then we compute a $(1 + \epsilon_0)\alpha$-approximation to the original capacitated k-median problem.

The strategy is similar to that in [11]: we guess a set of *leaders* and *distances* that match the optimal solution. More formally, let $F^* = \{f_1^*, \ldots, f_k^*\} \subseteq \mathbb{F}$ be the optimal solution with assignment μ^*. For each $f_i^* \in F^*$, let $(\mu^*)^{-1}(f_i^*)$ be the clients in the coreset assigned by μ^* to f_i^*, and let ℓ_i be the client in $(\mu^*)^{-1}(f_i^*)$ closest to f_i^*. We call ℓ_i the *leader* of the client set $(\mu^*)^{-1}(f_i^*)$. Also, let R_i be the distance $d(f_i^*, \ell_i)$, rounded down to the closest integer power of $(1 + \epsilon_1)$ for some ϵ_1 we set later.

The algorithm begins with an enumeration phase. There are $|W|^k$ choices for the set $\{\ell_1, \ldots, \ell_k\}$, and $O(\epsilon_1^{-1} \log n)^k$ choices for the values $R_1, \ldots, R_k$, since we assumed that the instance has aspect ratio $\mathrm{poly}(n)$. So by enumerating over $|W|^k O(\epsilon_1^{-1} \log n)^k = (k \log n\, \epsilon_0^{-1} \epsilon_1^{-1})^{O(k)}$ choices, we can assume that we have guessed the right values ℓ_i and R_i.

For each leader ℓ_i, define $\mathbb{F}_i$ as the centers $f \in \mathbb{F}$ satisfying $d(\ell_i, f) \in [1, 1 + \epsilon_1) \cdot R_i$. Note that $f_i^* \in \mathbb{F}_i$ for each i. Next, the algorithm wants to pick the center in each $\mathbb{F}_i$ with the largest capacity. This way, even if it doesn't pick f_i^* for $\mathbb{F}_i$, it picks a center not much farther away that has at least as much capacity.

The most natural solution is to *greedily* choose the center with largest capacity in each $\mathbb{F}_i$. One immediate issue with this approach is that we might choose the same center twice, since the sets $\mathbb{F}_i$ are not necessarily disjoint. Note that this issue is not as pronounced in the uncapacitated k-median problem, since in that case, we can always imagine choosing the same center twice and then throwing out one copy, which changes nothing. In the capacitated case, choosing the same center twice effectively doubles the capacity at that center, so throwing out a copy affects the capacity at that center.

One simple fix to this issue is the simple idea of *color-coding*, common in the FPT literature: for each center $f \in \mathbb{F}$, independently assign a uniformly random label in $\{1, 2, 3, \dots, k\}$. With probability $1/k^k$, each $f_i^* \in F^*$ is assigned label i. Moreover, repeating this routine $O(k^k \log n)$ times ensures that w.h.p., this will happen in some iteration. So with a $O(k^k \log n)$ multiplicative overhead in the running time, we may assume that each f_i^* is assigned label i.

The algorithm now chooses, from each $\mathbb{F}_i$, the center with the largest capacity among all centers with label i. Since f_i^* is an option for each $\mathbb{F}_i$, the center chosen can only have larger capacity. Let the center chosen from $\mathbb{F}_i$ be f_i. Let $F := \{f_1, \dots, f_k\}$ be our chosen centers.

We now claim that F is a $(3+\epsilon_1)$-approximation. Recall μ^*, the optimal assignment to the centers F^*; we construct an assignment μ to F as follows: for each client c in the coreset, if μ^* assigns c to center f_i^*, then we set $\mu(c) = f_i$. Observe that if $\mu^*(c) = f_i^*$, then

$$d(c, f_i) \le d(c, f_i^*) + d(f_i^*, \ell_i) + d(\ell_i, f_i) \le d(c, f_i^*) + 2(1+\epsilon_1)R_i \le d(c, f_i^*) + 2(1+\epsilon_1) \cdot d(c, f_i^*),$$

where the first inequality follows from triangle inequality, the second follows since both f_i^* and f_i are approximately R_i away from ℓ_i, and the third follows from $d(c, f_i^*) \ge d(\ell_i, f_i^*) \ge R$ by our choice of ℓ_i. Therefore, we have $d(c, \mu(c)) = d(c, f_i) \le (3+2\epsilon_1) d(c, f_i^*) = (3 + 2\epsilon_1) d(c, \mu^*(c))$. Altogether, the total cost of the assignment μ is

$$\sum_{(c,w)\in W} w \cdot d(c, \mu(c)) \le \sum_{(c,w)\in W} w \cdot (3+2\epsilon_1) d(c, \mu^*(c)) = (3+2\epsilon_1)\, OPT.$$

The optimal assignment can only be better, hence the $(3+2\epsilon_1)$-approximation. This implies a $(1+\epsilon_0)(3+2\epsilon_1)$-approximation in time $\text{poly}(k \log n\, \epsilon_0^{-1} \epsilon_1^{-1})^{O(k)}$. Finally, setting $\epsilon_0, \epsilon_1 := \Theta(\epsilon)$, for $\Theta(\cdot)$ small enough, guarantees a $(3+\epsilon)$-approximation in time $(k \log n\, \epsilon^{-1})^{O(k)} n^{O(1)}$.

Lastly, we show that the $(\log n)^{O(k)}$ factor in the running time can be upper bounded by $k^{O(k)} n^{O(1)}$, proving the second running time in Theorem 1. If $k < \frac{\log n}{\log\log n}$, then $(\log n)^{O(k)} = (\log n)^{\frac{\log n}{\log \log n}} = n^{O(1)}$; otherwise, $k > \frac{\log n}{\log \log n} \ge \sqrt{\log n}$, so $(\log n)^{O(k)} \le (k^2)^{O(k)}$. Therefore, the running time in Theorem 1 is at most $O(k/\epsilon)^{O(k)} n^{O(1)}$.

For k-means, the algorithm and analysis are identical, except that the total cost is now

$$_{(c,w)\in W} w \cdot d(c, \mu(c))^2 \le \sum_{(c,w)\in W} w \cdot \big((3+2\epsilon_1) d(c, \mu^*(c))\big)^2 = (9 + O(\epsilon_1))\, OPT,$$

implying a $(9+\epsilon)$-approximation. This concludes the proof of Theorem 1.

3 A $(1+\epsilon)$-Approximation for Euclidean Inputs

3.1 The Continuous (Uniform-Capacity) Case – Proof of Theorem 3

In this section we consider the continuous case: namely the case where centers can be located at arbitiary position in $\mathbb{R}^d$ and the capacities are uniform and $\eta \ge n/k$.

Let $\epsilon > 0$. Given a set of points P, denote by $\text{OPT}_1(P)$ the location of the optimal center of P (namely, the centroid of P in the case of the k-means problem or the median of P in the case of the k-median problem). We will make us of the following lemma of [20].

▶ **Lemma 21** (Lemma 5.3 in [20]). *Let P be a set of points in $\mathbb{R}^d$ and X be a random sample of size $O(\epsilon^{-3} \log(1/\epsilon))$ from P and a and b such that $a \le \mathsf{cost}(P, OPT_1(P)) \le b$. Then, we can construct a set Y of $O(2^{1/\epsilon^{O(1)}} \log(b/\epsilon a))$ points such that with constant probability there is at least one point $z \in X \cup Y$ satisfying $\mathsf{cost}(P, \{z\}) \le (1+2\epsilon)\mathsf{cost}(P, OPT_1(P))$. Further, the time taken to construct Y from X is $O(2^{1/\epsilon^{O(1)}} \log(b/\epsilon a) d)$.*

Our algorithm for obtaining a $(1+\epsilon)$-approximation is as follows:

1. Compute a coreset C for capacitated k-median as described by Lemma 21, and an estimate γ of the value of OPT using the classic $O(\log n)$-approximation.
   In the remaining, we assume that the minimum pairwise distance between pairs of points of C is at least $\epsilon\gamma/(n \log n)$ since otherwise one can simply take a net of the input and the additive error is at most ϵOPT (see e.g.: [11]). Moreover, we assume that there is no cluster containing only one point of the coreset since these clusters can be "guessed" and dealt with separately.
2. Start with $\mathcal{C} = \emptyset$, then for each subset S of C of size $O(\epsilon^{-3}\log(k/\epsilon))$, for each $s = (1+\epsilon)^i$ in the interval $[\epsilon\gamma/(n\log n), \gamma]$ apply the procedure of Lemma 21 with $a = s$ and $b = (1+\epsilon)a$ and add the output of the procedure to $\mathcal{C}$. We refer to $\mathcal{C}$ as a set of approximate candidate centers.
3. Consider all subsets of size k of $\mathcal{C}$. For each subset, compute the cost of using this set of centers for the capacitated k-median instance by using a min cost flow computation. Output the set of centers of minimum cost.

We first discuss the running time of the algorithm. The time for computing the coreset is polynomial by Theorem 11. Generating $\mathcal{C}$ takes $|C|^{O(\epsilon^{-3}\log(1/\epsilon))} \cdot 2^{1/\epsilon^{O(1)}} \log((1+\epsilon)/\epsilon)d$ time. For the last part, namely enumerating all subsets of $\mathcal{C}$ of size k, the running time is $|C|^{O(k\epsilon^{-3}\log(1/\epsilon))} \cdot 2^{k/\epsilon^{O(1)}} \log^k((1+\epsilon)/\epsilon)$. Theorem 11 implies that $|C| = \text{poly}(k \log n\, \epsilon^{-1})$ and so, the algorithm has running time $(k\log n\, \epsilon^{-1})^{k\epsilon^{-O(1)}} n^{O(1)}$. Again, the $(\log n)^{k\epsilon^{-O(1)}}$ factor can be upper bounded by $(k/\epsilon)^{k\epsilon^{-O(1)}}$ or $n^{O(1)}$ based on whether or not $k\epsilon^{-O(1)} < \frac{\log n}{\log\log n}$, hence the improved running time in Theorem 3.

We show that this algorithm provides a $(1+O(\epsilon))$-approximation. Theorem 11 immediately implies that the solution found for the coreset C can be lifted to a solution for the original input at a cost of an additive $O(\epsilon\text{OPT})$. For any (possibly weighted) set of client A and set of centers B, we define $\mathsf{cost}(A, B)$ to be the cost of the best assignment of the clients in A to the centers of B.

▶ **Lemma 22.** *The $\mathcal{C}$ computed by the algorithm contains a set of centers $\tilde{S}$ that is such that* $\mathsf{cost}(C, \tilde{S}) \le (1+\epsilon)\mathsf{cost}(C, OPT)$.

Proof. This follows almost immediately from Lemma 21. By Lemma 21, for each cluster C_i^* of OPT, there exists a set $S_i^* \subseteq C_i^*$ of size at most $O(\epsilon^{-3}\log(k/\epsilon))$ such that applying the procedure of Lemma 21 with the correct value of a to S_i^* yields a set of points containing a point z_i such that $\mathsf{cost}(C_i^*, z_i) \le (1+2\epsilon)\mathsf{cost}(C_i^*, \text{OPT})$. Since the algorithm iterates over all subsets of size $O(\epsilon^{-3}\log(k/\epsilon))$, and that the pairwise distance is at least $\epsilon\text{OPT}/n$, it follows that S_i^* is one of the subset considered by the algorithm, and so z_i is part of $\mathcal{C}$. ◀

Finally, since the algorithm iterates over all subsets of $\mathcal{C}$ of size at most k, Lemma 22 implies that there exists a set $\{z_1, \ldots, z_k\}$ that is considered by the algorithm and on which solving a min cost flow instance yields a solution of cost at most $(1+O(\epsilon))\mathsf{cost}(\mathcal{P}, \text{OPT})$.

3.2 The Non-Uniform Case – Proof of Theorem 2

We now consider the non-uniform case. In this setting, the input consists of a set of points in $\mathbb{R}^d$ together with a set of candidate centers in $\mathbb{R}^d$ and a capacity η_f for each such candidate center. We make use of the following lemma. As slightly worse bound for the lemma can also be found in [25].

▶ **Lemma 23** ([27]). *Let $\epsilon \in (0,1)$ and $X \subseteq \mathbb{R}^d$ be arbitrary with X having size $n > 1$. There exists $f : \mathbb{R}^d \mapsto \mathbb{R}^m$ with $m = O(\epsilon^{-2} \log n)$ such that $\forall x \in X$, $\forall y \in \mathbb{R}^d$, $||x - y||_2 \leq ||f(x) - f(y)||_2 \leq (1+\epsilon)||x - y||_2$.*

We describe a polynomial-time approximation scheme. Let $\epsilon > 0$. The algorithm is as follows. The first step of the algorithm is identical to the continous case.

1. Compute a coreset C for capacitated k-median as described by Theorem 21, and an estimate γ of the value of OPT using the classic $O(\log n)$-approximation.

   In the remaining, we assume that the minimum pairwise distance between pairs of points of C is at least $\epsilon\gamma/(n \log n)$ since otherwise one can simply take a net of the input and the additive error is at most ϵOPT (see e.g.: [11]). Moreover, we assume that there is no cluster containing only one point of the coreset since these clusters can be "guessed" and dealt with separately.
2. Apply Lemma 23 to the points of the coreset to obtain a set of points in a Euclidean space of dimension $\frac{\log k + \log\log n}{\epsilon^{O(1)}}$. Let C^* and A^* be respectively the image of the coreset points and of the candidate centers through the projection.
3. Start with $\mathcal{V} = \emptyset$ For each point p of the coreset do the following: For each $i \in \{1, 2, \dots, n^2\}$, consider the ith-*ring* defined by $\mathsf{ball}(p, (1+\epsilon)^i \epsilon\gamma/(n \log n)) \setminus \mathsf{ball}(p, (1+\epsilon)^{i-1}\epsilon\gamma/(n \log n))$ and choose an $\epsilon \cdot (1+\epsilon)^i \epsilon\gamma/(n \log n)$-net. Consider the Voronoi diagram induced by the points of the net. Then, for each Voronoi cell, add to $\mathcal{V}$ the k candidate centers of A^* in the cell that are of maximum capacity.
4. Enumerate all possible subset of $\mathcal{V}$ of size k and output the one that leads to the solution of minimum cost.

3.2.1 Correctness

Theorem 11 implies that finding a near-optimal solution for the coreset points yields a near-optimal solution for the input point set.

Lemma 23 immediately implies that, given the coreset construction C, and the projection of the coreset points onto a $\frac{\log k + \log\log n}{\epsilon^{O(1)}}$-dimensional Euclidean space, finding a near-optimal set of centers in A^* yields a near-optimal set of centers in A through the inverse of the projection.

Therefore, it remains to show that the set $\mathcal{V}$ contains a set of candidate centers that yields a near-optimal solution. To see this, consider each center of the optimal solution in A^*. For each such optimal center f, consider the closest coreset point $c(f)$ together with the ring of $c(f)$ containing f. Let j be the index of this ring, namely $f \in \mathsf{ball}(p, (1+\epsilon)^j \epsilon\gamma/(n \log n)) \setminus \mathsf{ball}(p, (1+\epsilon)^{j-1}\epsilon\gamma/(n \log n))$.

By definition of the net, there exists a point p of the net at distance at most $\epsilon \cdot \mathsf{ball}(p, (1+\epsilon)^j \epsilon\gamma/(n \log n)) \leq 2\epsilon||c - c(f)||_2$ from $c(f)$. Therefore, consider the Voronoi cell of p and the top-k candidate centers in terms of capacity. If f is part of this top-k, then f is part of $\mathcal{V}$ and we are done. Otherwise, it is possible to associate to f a center f^* that has capacity at least the capacity of f, and so for all the optimal centers simultaneously since we consider the top-k. Therefore, consider replacing f by f^* in the optimal solution. The change in cost is at most, by the triangle inequality, $4\epsilon||c - c(f)||_2$ since both centers are in the Voronoi cell of p. Finally, since c is the closest client to $c(f)$, the cost increases by a factor at most $(1 + 4\epsilon)$ for each client and the correctness follows.

3.2.2 Running time

We now bound the running time. The first two steps are clearly polynomial time. An $\epsilon \cdot (1+\epsilon)^i \epsilon\gamma/(n \log n)$-net of a ball of radius $(1+\epsilon)^i \epsilon\gamma/(n \log n)$ has size $\epsilon^{-O(d)}$ and so in this context, after Step 2, a size $\epsilon^{-(\frac{\log k + \log\log n}{\epsilon^{O(1)}})}$. Since for each element of the net, k centers are chosen and since the number of rings is, by Step 1, at most $O(\epsilon^{-2} \log n)$, the total size of $\mathcal{V}$ is at most $|C| k \epsilon^{-2} \log n \epsilon^{-(\frac{\log k + \log\log n}{\epsilon^{O(1)}})}$ which is at most $|C|\epsilon^{-2}(k \log n)^{\epsilon^{-O(1)}} = (k\epsilon^{-1} \log n)^{\epsilon^{-O(1)}}$. Enumerating all subsets of size k takes time $(k\epsilon^{-1} \log n)^{k\epsilon^{-O(1)}}$ and the theorem follows.

References

1. M. Adamczyk, J. Byrka, J. Marcinkowski, S. M. Meesum, and M. Włodarczyk. Constant factor FPT approximation for capacitated k-median. *ArXiv e-prints*, September 2018. arXiv:1809.05791.
2. 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 the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 106–113, 1998. doi:10.1145/276698.276718.
3. Jarosław Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 722–736. SIAM, 2014.
4. Jarosław Byrka, Bartosz Rybicki, and Sumedha Uniyal. An Approximation Algorithm for Uniform Capacitated k-Median Problem with $1+\epsilon$ Capacity Violation. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 262–274. Springer, 2016.
5. Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-Median. In *STOC*, volume 98, pages 114–123. Citeseer, 1998.
6. Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.
7. K. Chen. On Coresets for k-Median and k-Means Clustering in Metric and ELuclidean Spaces and Their Applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.
8. Julia Chuzhoy and Yuval Rabani. Approximating K-median with Non-uniform Capacities. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 952–958, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=1070432.1070569.
9. Vincent Cohen-Addad. Approximation Schemes for Capacitated Clustering in Doubling Metrics. *CoRR*, abs/1812.07721, 2018. arXiv:1812.07721.
10. Vincent Cohen-Addad, Arnaud de Mesmay, Eva Rotenberg, and Alan Roytman. The Bane of Low-Dimensionality Clustering. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 441–456, 2018. doi:10.1137/1.9781611975031.30.
11. Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In *ICALP 2019*, 2019.
12. Wenceslas Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 50–58. ACM, 2003. doi:10.1145/780542.780550.
13. H. Gökalp Demirci and Shi Li. Constant Approximation for Capacitated k-Median with (1+epsilon)-Capacity Violation. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.73.

14 G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *STOC*, pages 209–217, 2005.

15 Anupam Gupta, Euiwoong Lee, and Jason Li. An FPT Algorithm Beating 2-approximation for K-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 2821–2837, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3174304.3175483.

16 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michal Wlodarczyk. Losing Treewidth by Separating Subsets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1731–1749, 2019. doi:10.1137/1.9781611975482.104.

17 Sariel Har-Peled and Akash Kushal. Smaller Coresets for k-Median and k-Means Clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007. doi:10.1007/s00454-006-1271-x.

18 Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 291–300, 2004. doi:10.1145/1007352.1007400.

19 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A Simple Linear Time (1+ ") - Approximation Algorithm for k-Means Clustering in Any Dimensions. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 454–462, Washington, DC, USA, 2004. IEEE Computer Society. doi:10.1109/FOCS.2004.7.

20 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2), 2010. doi:10.1145/1667053.1667054.

21 Euiwoong Lee. Partitioning a graph into small pieces with applications to path transversal. *Mathematical Programming*, March 2018. doi:10.1007/s10107-018-1255-7.

22 Shi Li. On Uniform Capacitated k-Median Beyond the Natural LP Relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015. doi:10.1137/1.9781611973730.47.

23 Shi Li. Approximating capacitated k-median with $(1 + k$ open facilities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 786–796, 2016. doi:10.1137/1.9781611974331.ch56.

24 Shi Li. On Uniform Capacitated k-Median Beyond the Natural LP Relaxation. *ACM Trans. Algorithms*, 13(2):22:1–22:18, 2017. doi:10.1145/2983633.

25 Sepideh Mahabadi, Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Nonlinear Dimension Reduction via Outer Bi-Lipschitz Extensions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 1088–1101, New York, NY, USA, 2018. ACM. doi:10.1145/3188745.3188828.

26 Dániel Marx and Michal Pilipczuk. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 865–877, 2015. doi:10.1007/978-3-662-48350-3_72.

27 Shyam Narayanan and Jelani Nelson. Optimal terminal dimensionality reduction in Euclidean space. *CoRR – To appear in the proceedings of STOC'19*, abs/1810.09250, 2018. arXiv:1810.09250.

28 Yicheng Xu, Yong Zhang, and Yifei Zou. A constant parameterized approximation for hard-capacitated k-means. *CoRR*, abs/1901.04628, 2019. arXiv:1901.04628.