



HAL
open science

On the Fixed-Parameter Tractability of Capacitated Clustering

Vincent Cohen-Addad, Jason Li

► **To cite this version:**

Vincent Cohen-Addad, Jason Li. On the Fixed-Parameter Tractability of Capacitated Clustering. 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019), Jul 2019, Patras, Greece. pp.41:1–41:14, 10.4230/LIPIcs.ICALP.2019.41 . hal-02169579

HAL Id: hal-02169579

<https://hal.science/hal-02169579>

Submitted on 1 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Fixed-Parameter Tractability of Capacitated Clustering

Vincent Cohen-Addad

CNRS & Sorbonne Université

Jason Li

Carnegie Mellon University

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 \rightarrow Facility location and clustering; Theory of computation \rightarrow Fixed parameter tractability; Mathematics of computing \rightarrow Probabilistic algorithms; Mathematics of computing \rightarrow 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.CVIT.2016.23

Funding 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.*: [5, 23, 24, 25, 14, 4, 9, 7]), 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 [6] 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 [23, 24] or the number of clusters can be violated by a $(1 + \epsilon)$ factor [5, 14], 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.

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



© Vincent Cohen-Addad and Jason Li;
licensed under Creative Commons License CC-BY
42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:15



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

45 (QPTAS) for capacitated k -median in Euclidean space of fixed dimension with $(1+\epsilon)$ capacity
 46 violation was known since the late 90's [3]. This has been recently improved to a PTAS
 47 for \mathbb{R}^2 and a QPTAS for doubling metrics without capacity violation [10]. It remains an
 48 interesting open question to obtain constant factor approximation for other metrics such as
 49 planar graphs or Euclidean space of arbitrary dimension.

50 For many optimization problems are at least W[1]-hard and so obtaining exact fixed-
 51 parameter tractable (FPT) algorithms is unlikely. However, FPT algorithms have recently
 52 shown that they can help break long-standing barriers in the world of approximation
 53 algorithms. FPT approximation algorithms achieving better approximation guarantees than
 54 the best known polynomial-time approximation algorithms for some classic W[1]- and W[2]-
 55 hard problems have been designed. For example, for k -cut [16], for k -vertex separator [22] or
 56 k -treewidth-deletion [17].

57 For the fixed-parameter tractability of the k -median and k -means problems, a natural
 58 parameter is the number of clusters k . The FPT complexity of the classic uncapacitated
 59 k -median problem, parameterized by k , has received a lot of attention over the last 15 years.
 60 From a lower bound perspective, the problem is known to be W[2]-hard in general metric
 61 spaces and assuming the exponential time hypothesis (ETH), even for points in \mathbb{R}^4 , there is
 62 no exact algorithm running in time $n^{o(k)}$ [11]. For \mathbb{R}^2 there exists an exact $n^{O(\sqrt{k})}$ which is
 63 the best one can hope for assuming ETH [11], see also [27].

64 From an upper bound perspective, *coreset* constructions and PTAS with running time
 65 $f(k, \epsilon)n^{O(1)}$ have been known since the early 00's [13, 20, 18, 19, 15]. In the language of
 66 fixed-parameter tractability, a coreset is essentially an “approximate kernel” for the problem:
 67 given a set P of n points in a metric space, a coreset is, loosely speaking, a mapping from
 68 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
 69 cost γ can be converted into a clustering of P of cost at most $\gamma \pm \epsilon \text{cost}(\text{OPT})$, through the
 70 inverse of the mapping (where OPT is the optimal solution for P). See Definition 9 for a
 71 more complete definition.

72 In Euclidean space, several coreset constructions for uncapacitated k -median are inde-
 73 pendent of the input size and of the dimension and so are truly approximate kernels. Thus
 74 approximation schemes can simply be obtained by enumerating all possible partitions of
 75 the coreset points into k parts, evaluating the cost of each of them and outputting the one
 76 of minimum cost. However, obtaining similar results in general metric spaces seems much
 77 harder and is likely impossible. In fact, obtaining an FPT approximation algorithm with
 78 approximation guarantee less than $1 + 2/e$ is impossible assuming Gap-ETH, see [12].

79 For the capacitated k -median and k -means problems much less is known. First, the
 80 coreset constructions or the classic FPT-approximation schemes techniques of [21, 13] do not
 81 immediately apply. Thus, very little was known until the recent result of Adamczyk et al. [1]
 82 who proposed a $(7 + \epsilon)$ -approximation algorithm running in time $k^{O(k)}n^{O(1)}$. More recently,
 83 a $(69 + \epsilon)$ -approximation algorithm for the capacitated k -means problem with similar running
 84 time has been proposed by Xu et al. [29].

85 1.1 Our Results

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

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

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

95 This results in a significant improvement over the recent results of Adamczyk et al. [1] for
 96 k -median and Xu et al. [29] for (Euclidean) k -means, in the same asymptotic running time.

97 Moreover, combining with the techniques of Kumar et al. [21], we obtain a $(1 + \epsilon)$ -
 98 approximation algorithm for points in \mathbb{R}^d , where d is arbitrary. We believe that this is an
 99 interesting result: while it seems unlikely that one can obtain an FPT-approximation better
 100 than $1 + 2/e$ in general metrics, it is possible to obtain an FPT- $(1 + \epsilon)$ -approximation in
 101 Euclidean metrics of arbitrary dimension. This works for both the *discrete* and *continuous*
 102 settings: in the former, the set of centers must be chosen from a discrete set of candidate
 103 centers in \mathbb{R}^d and the capacities may not be uniform, while in the latter the centers can be
 104 placed anywhere in \mathbb{R}^d and the capacities are uniform.

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

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

111 These two results are a major improvement over the 69-approximation algorithm of Xu
 112 et al. [29].

113 1.2 Preliminaries

114 We now provide a more formal definition of the problems.

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

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

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

124 In the case of the capacitated Euclidean k -median and k -means, our approach works for
 125 the two main definitions. First, the definition of [29, 21]: $P = \mathbb{R}^d$ and capacities are uniform,
 126 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
 127 $f \in P$, the input specifies a specific capacity η_f

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

23:4 On the Fixed-Parameter Tractability of Capacitated Clustering

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

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

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

141 ► **Definition 8.** *We define CapKMeans(C, F) and FracCapKMeans(W, F) similarly, except*
 142 *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.*

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

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

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

$$155 \quad \text{FracCapKMed}(W, F) \in (1 - \epsilon, 1 + \epsilon) \cdot \text{CapKMed}(C, F).$$

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

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

$$160 \quad \min_{\substack{F \subseteq \mathbb{F} \\ |F|=k}} \text{FracCapKMed}(W, F) \in (1 - \epsilon, 1 + \epsilon) \cdot \min_{\substack{F \subseteq \mathbb{F} \\ |F|=k}} \text{CapKMed}(C, F),$$

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

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

¹ For example, the following modification to the distances d does the trick. First, compute an $O(\log k)$ -
 approximation [6] 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 $O(n^{20})$.

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

173 2 Coreset for k -median

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

176 ► **Theorem 11.** *For any small enough constant $\epsilon \geq 0$, there exists a Monte Carlo algorithm
 177 that, given an instance $((V, d), C, \mathbb{F}, k)$ of capacitated k -median, outputs a (strong) coreset
 178 $W \subseteq C$ with size $O(k^2 \log^2 n / \epsilon^3)$ in polynomial time, w.h.p.*

179 ► **Theorem 12.** *For any small enough constant $\epsilon \geq 0$, there exists a Monte Carlo algorithm
 180 that, given an instance $((V, d), C, \mathbb{F}, k)$ of capacitated k -means, outputs a (strong) coreset
 181 $W \subseteq C$ with size $O(k^5 \log^5 n / \epsilon^3)$ in polynomial time, w.h.p.*

182 Our inspiration for the coreset construction is Chen’s algorithm [8] based on random
 183 sampling. Our algorithm is essentially the same, with slightly worse bounds in the sampling
 184 step, although our analysis is a lot more involved. We describe the full algorithm in
 185 pseudocode below (see Algorithm 1).

186 At a high level, the algorithm first partitions the client set C into $\text{poly}(k, \log n)$ many
 187 subsets, called *rings*, with the help of a polynomial-time approximate solution (see line 1).
 188 The sets are called rings because they are of the form $C_i \cap (\text{ball}(f'_i, R) \setminus \text{ball}(f'_i, R/2))$ for
 189 some subset of clients $C_i \subseteq C$, some facility $f'_i \in \mathbb{F}$, and some positive number R (see
 190 line 7). Then, for each ring $C_{i,R}$, if $|C_{i,R}|$ is small enough, the algorithm adds the entire ring
 191 into the coreset (each with weight 1); otherwise, the algorithm takes a random sample of
 192 $r = \text{poly}(k, \log n)$ many clients in $C_{i,R}$, weights each sampled client by $|C_{i,R}|/r$, and adds the
 193 weighted sample to the coreset. The weighting ensures that the total weight of the sampled
 194 points is always equal to $|C_{i,R}|$. To prove that the algorithm produces a coreset w.h.p., Chen
 195 union bounds over all $\binom{|\mathbb{F}|}{k}$ choices of a set of k facilities, and shows that for each choice
 196 $F \subseteq \mathbb{F}$, with probability at least $1 - n^{-\Omega(k)}$, the total cost to assign the coreset points to F
 197 is approximately the total cost to assign the original clients C to F ; this statement is proved
 198 through standard concentration bounds. More details and intuition for the algorithm can be
 199 found in Section 3 of Chen’s paper [8].

200 2.1 Single ring case

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

206 Fix an arbitrary ring $C_{i,R}$ throughout this section, and define $C' := C_{i,R}$ for convenience.
 207 Let $N := |C'|$ be the number of clients, and let $f' := f'_i$ be the ring center of C' (line 4).
 208 Let W' be the (weighted) centers in $C_{i,R}$ sampled by the algorithm (lines 12–13), together
 209 with the (unweighted) centers in $C \setminus C'$, which have weight 1. Our goal is to show that
 210 $\text{FracCapKMed}(W', F)$, the cost after sampling only from C' , is close to the original cost
 211 $\text{CapKMed}(C, F)$.

Algorithm 1 CoreSet(I)

1: $F' = \{f'_1, \dots, f'_{O(k)}\} \leftarrow$ an $(O(1), O(1))$ bicriteria solution to instance I , namely a capacitated $O(k)$ -median solution with total cost $ALG' \leq O(OPT)$ \triangleright using, e.g., [24]
 2: $W \leftarrow \emptyset$ $\triangleright W \subseteq C \times \mathbb{R}_+$ 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 $C \cup \mathbb{F}$ $\triangleright d_{\max}/d_{\min}$ is the aspect ratio
 4: **for** each center f'_i **do** $\triangleright O(k)$ 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** $\triangleright O(\log n)$ iterations, assuming $\text{poly}(n)$ aspect ratio
 7: $C_{i,R} \leftarrow C_i \cap (\text{ball}(f'_i, R) \setminus \text{ball}(f'_i, R/2))$ \triangleright 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 $c \in C_{i,R}$ $\triangleright C_{i,R}$ 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 \triangleright weighted so that total weight is still $|C_{i,R}|$

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

$$213 \quad \left| \text{FracCapKMed}(W', F) - \text{CapKMed}(C, F) \right| \leq \epsilon NR. \quad (1)$$

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

217 The intuition behind the ϵNR additive error is that we can “charge” this error to the
 218 cost of the bicriteria solution (line 1) that C' is responsible for. In particular, the total cost
 219 of assigning clients in C' to ring center f' in the bicriteria solution is at least $N \cdot R/2$, since
 220 all clients in C' are distance at least $R/2$ to f' . Therefore, we charge an additive error of
 221 ϵNR to a $NR/2$ portion of ALG' , which is a “rate” of 2ϵ to 1. If we can do the same for
 222 all rings, then since the portions of ALG' sum to ALG' , our total additive error is at most
 223 $2\epsilon \cdot ALG' = O(\epsilon) \cdot OPT$. Finally, replacing ϵ with a small enough $\Theta(\epsilon)$ gives the desired
 224 additive error of $\epsilon \cdot OPT$; note that this is where we use that the approximation ratio of
 225 ALG' is $O(1)$, and that the specific approximation ratio is not important (as long as it is
 226 constant). The formalization of this intuition is deferred to the full version; the argument is
 227 identical to Chen’s [8], so we claim no novelty here.

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

231 Our high-level strategy is the same as Chen’s: we union bound over all sets of centers
 232 $F \subseteq \mathbb{F}$ of size k , and prove that for a fixed set F , the probability of violating (1) is at most
 233 $n^{-(k+10)}$.² Union bounding over all $\leq \binom{n}{k}$ choices of F gives probability $\leq n^{-10}$ of violating

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 probability $1 - n^{-Z}$ for any positive constant Z .

234 (1), proving the lemma. Therefore, from now on, we focus on a single, arbitrary set $F \subseteq \mathbb{F}$
 235 of size k satisfying $\text{CapKMed}(C, F) < \infty$, and aim to show that (1) fails with probability
 236 $\leq n^{-(k+10)}$.

237 For our analysis, we define a function $g : \mathbb{R}_+^{C'} \rightarrow \mathbb{R}_+$ as follows. For an input vector
 238 $\mathbf{d} \in \mathbb{R}_+^{C'}$ (indexed by clients in C'), consider a min-cost flow instance $\text{FlowInstance}(\mathbf{d})$ on the
 239 graph metric with the following demands: set demand d_c at each client $c \in C'$, demand 1
 240 at each client $c \in C \setminus C'$, and demand $N - \sum_{c \in C'} d_c$ (this demand can be negative) at ring
 241 center $f' = f'_i$ (so we are effectively treating f' as a special client with possibly negative
 242 demand, not a facility). Observe that $\text{FlowInstance}(\mathbf{d})$ is a feasible min-cost flow instance,
 243 because the sum of demands is exactly

$$244 \quad \sum_{c \in C'} d_c + |C \setminus C'| + \left(N - \sum_{c \in C'} d_c \right) = |C \setminus C'| + N = |C|,$$

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

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

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

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

259 In light of all this, our main argument has two steps. First, we show that $g(X)$ is
 260 concentrated around $\mathbb{E}[g(X)]$ using martingales. However, what we really need is con-
 261 centration around $g(\mathbb{E}[X]) = g(\mathbf{1}) = \text{CapKMed}(C, F)$, so our second step is to show that
 262 $\mathbb{E}[g(X)] \approx g(\mathbb{E}[X])$ (with probability 1). We formally state the lemmas below which, as
 263 discussed, together imply Lemma 13.

264 \blacktriangleright **Lemma 15.** Assume that $|C'| > \Theta(k \log n / \epsilon^3)$. With probability $\geq 1 - n^{-(k+20)}$, we have
 265 $|g(X) - \mathbb{E}[g(X)]| \leq \epsilon NR/2$.

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

267 2.1.1 Proof of Lemma 15: concentration around $\mathbb{E}[g(X)]$ via 268 martingales.

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

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

272 **Proof.** Fix a client $c \in C'$, and consider two vectors $\mathbf{d}, \mathbf{d}' \in \mathbb{R}_+^{C'}$ with $\mathbf{d}' = \mathbf{d} + \delta \cdot \mathbf{1}_c$. By
 273 definition of FlowInstance , the only difference between $\text{FlowInstance}(\mathbf{d})$ and $\text{FlowInstance}(\mathbf{d}')$
 274 is that in $\text{FlowInstance}(\mathbf{d}')$, client c has δ more demand and “special client” f' has δ less

23:8 On the Fixed-Parameter Tractability of Capacitated Clustering

275 demand. Therefore, if we begin with the min-cost flow of $\text{FlowInstance}(\mathbf{d})$, and then add
 276 δ units of flow from c to f' , then we now have a feasible flow for $\text{FlowInstance}(\mathbf{d}')$.³ This
 277 means that

$$278 \quad g(\mathbf{d}') \leq g(\mathbf{d}) + \delta R.$$

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

$$281 \quad g(\mathbf{d}) \leq g(\mathbf{d}') + \delta R.$$

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

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

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

$$288 \quad \Pr [|g(X) - \mathbb{E}[g(X)]| \geq \epsilon p n b L] \leq 2e^{-\epsilon^2 p n / 3}$$

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

$$\begin{aligned} 292 \quad & \Pr [|g(X) - \mathbb{E}[g(X)]| \geq (\epsilon/2)NR] \\ 293 \quad &= \Pr [|g(X) - \mathbb{E}[g(X)]| \geq (\epsilon/2)pnbL] \\ 294 \quad &\leq 2 \exp\left(\frac{-(\epsilon/2)^2 p n}{3}\right) \\ 295 \quad &= 2 \exp\left(\frac{-(\epsilon/2)^2 (r/N)N}{3}\right) = \exp(-\Theta(\epsilon^2 r)) = \exp\left(-\Omega(\epsilon^2 \cdot \frac{k \log n}{\epsilon^2})\right) \\ 296 \quad &\leq n^{-(k+20)} \end{aligned}$$

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

299 2.1.2 Proof of Lemma 16: relating $\mathbb{E}[g(X)]$ with $g(\mathbb{E}[X])$.

300 We have obtained concentration about $\mathbb{E}[g(X)]$, but we really need concentration around
 301 $g(\mathbb{E}[X]) = \text{CapKMed}(C', F)$. We establish this by proving Lemma 16.

302 We first show the easy direction, that $g(\mathbb{E}[X]) \leq \mathbb{E}[g(X)]$, which essentially follows from
 303 the convexity of min-cost flow: Suppose the outcomes of random variable X are $\mathbf{d}_1, \mathbf{d}_2, \dots$
 304 with respective probabilities μ_1, μ_2, \dots , so that $\mathbb{E}[g(X)] = \sum_i \mu_i g(\mathbf{d}_i)$. Now consider the
 305 flow obtained by adding up, for each i , the min-cost flow of $\text{FlowInstance}(\mathbf{d}_i)$ scaled by μ_i .
 306 This flow is a feasible flow to $\text{FlowInstance}(\mathbb{E}[X])$ and has cost at most $\mathbb{E}[g(X)]$. Since the
 307 min-cost flow of $\text{FlowInstance}(\mathbb{E}[X])$ can only be lower, we have $g(\mathbb{E}[X]) \leq \mathbb{E}[g(X)]$.

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

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 .

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

310 **Proof.** Since $X \in [0, N/r]^N$, and since g is R -Lipschitz, the entire range of $g(X)$ is contained
 311 in some interval of length $N \cdot N/r \cdot R \leq N \cdot n \cdot R$. Since $\mathbb{E}[X] \in [0, N/r]^N$ as well, the value
 312 $g(\mathbb{E}[X])$ is also contained in that interval. The statement follows. ◀

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

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

$$\begin{aligned} 317 \quad \mathbb{E}[g(X)] &\leq n^{-10} \cdot (g(\mathbb{E}[X]) + nNR) + (1 - n^{-10})(g(\mathbb{E}[X]) + 0.49\epsilon NR) \\ 318 &= g(\mathbb{E}[X]) + (n^{-10} \cdot n + (1 - n^{-10}) \cdot 0.49\epsilon) NR \\ 319 &\leq g(\mathbb{E}[X]) + (\epsilon/2) NR, \end{aligned}$$

320 finishing the proof of Lemma 16.

322 2.2 $(3 + \epsilon)$ - and $(9 + \epsilon)$ -approximation – Proof of Theorem 1

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

325 Suppose we run the coresets for the capacitated k -median instance with parameter ϵ_0 (to
 326 be set later), obtaining a coresets $W \subseteq C \times \mathbb{R}^+$ of size $\text{poly}(k \log n \epsilon_0^{-1})$. We now want to
 327 compute some $F \subseteq \mathbb{F}$ of size k and an assignment μ of the clients in W to F minimizing
 328 $\sum_{(c,w) \in W} w \cdot d(c, \mu(c))$. By definition of coresets, if we compute an α -approximation to this
 329 problem, then we compute a $(1 + \epsilon_0)\alpha$ -approximation to the original capacitated k -median
 330 problem.

331 The strategy is similar to that in [12]: we guess a set of *leaders* and *distances* that match
 332 the optimal solution. More formally, let $F^* = \{f_1^*, \dots, f_k^*\} \subseteq \mathbb{F}$ be the optimal solution with
 333 assignment μ^* . For each $f_i^* \in F^*$, let $(\mu^*)^{-1}(f_i^*)$ be the clients in the coresets assigned by μ^*
 334 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
 335 set $(\mu^*)^{-1}(f_i^*)$. Also, let R_i be the distance $d(f_i^*, \ell_i)$, rounded down to the closest integer
 336 power of $(1 + \epsilon_1)$ for some ϵ_1 we set later.

337 The algorithm begins with an enumeration phase. There are $|W|^k$ choices for the
 338 set $\{\ell_1, \dots, \ell_k\}$, and $O(\epsilon_1^{-1} \log n)^k$ choices for the values R_1, \dots, R_k , since we assumed
 339 that the instance has aspect ratio $\text{poly}(n)$. So by enumerating over $|W|^k O(\epsilon_1^{-1} \log n)^k =$
 340 $(k \log n \epsilon_0^{-1} \epsilon_1^{-1})^{O(k)}$ choices, we can assume that we have guessed the right values ℓ_i and R_i .

341 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
 342 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
 343 largest capacity. This way, even if it doesn't pick f_i^* for \mathbb{F}_i , it picks a center not much farther
 344 away that has at least as much capacity.

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

352 One simple fix to this issue is the simple idea of *color-coding*, common in the FPT literature:
 353 for each center $f \in \mathbb{F}$, independently assign a uniformly random label in $\{1, 2, 3, \dots, k\}$.

23:10 On the Fixed-Parameter Tractability of Capacitated Clustering

354 With probability $1/k^k$, each $f_i^* \in F^*$ is assigned label i . Moreover, repeating this routine
 355 $O(k^k \log n)$ times ensures that w.h.p., this will happen in some iteration. So with a $O(k^k \log n)$
 356 multiplicative overhead in the running time, we may assume that each f_i^* is assigned label i .

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

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

$$363 \quad d(c, f_i) \leq d(c, f_i^*) + d(f_i^*, \ell_i) + d(\ell_i, f_i) \leq d(c, f_i^*) + 2(1 + \epsilon_1)R_i \leq d(c, f_i^*) + 2(1 + \epsilon_1) \cdot d(c, f_i^*),$$

364 where the first inequality follows from triangle inequality, the second follows since both f_i^*
 365 and f_i are approximately R_i away from ℓ_i , and the third follows from $d(c, f_i^*) \geq d(\ell_i, f_i^*) \geq R$
 366 by our choice of ℓ_i . Therefore, we have $d(c, \mu(c)) = d(c, f_i) \leq (3 + 2\epsilon_1)d(c, f_i^*) = (3 +$
 367 $2\epsilon_1)d(c, \mu^*(c))$. Altogether, the total cost of the assignment μ is

$$368 \quad \sum_{(c,w) \in W} w \cdot d(c, \mu(c)) \leq \sum_{(c,w) \in W} w \cdot (3 + 2\epsilon_1)d(c, \mu^*(c)) = (3 + 2\epsilon_1) OPT.$$

369 The optimal assignment can only be better, hence the $(3 + 2\epsilon_1)$ -approximation. This implies a
 370 $(1 + \epsilon_0)(3 + 2\epsilon_1)$ -approximation in time $\text{poly}(k \log n \epsilon_0^{-1} \epsilon_1^{-1})^{O(k)}$. Finally, setting $\epsilon_0, \epsilon_1 := \Theta(\epsilon)$,
 371 for $\Theta(\cdot)$ small enough, guarantees a $(3 + \epsilon)$ -approximation in time $(k \log n \epsilon^{-1})^{O(k)} n^{O(1)}$.

372 Lastly, we show that the $(\log n)^{O(k)}$ factor in the running time can be upper bounded by
 373 $k^{O(k)} n^{O(1)}$, proving the second running time in Theorem 1. If $k < \frac{\log n}{\log \log n}$, then $(\log n)^{O(k)} =$
 374 $(\log n)^{\frac{\log n}{\log \log n}} = n^{O(1)}$; otherwise, $k > \frac{\log n}{\log \log n} \geq \sqrt{\log n}$, so $(\log n)^{O(k)} \leq (k^2)^{O(k)}$. Therefore,
 375 the running time in Theorem 1 is at most $O(k/\epsilon)^{O(k)} n^{O(1)}$.

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

$$377 \quad \sum_{(c,w) \in W} w \cdot d(c, \mu(c))^2 \leq \sum_{(c,w) \in W} w \cdot ((3 + 2\epsilon_1)d(c, \mu^*(c)))^2 = (9 + O(\epsilon_1)) OPT,$$

378 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

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

383 Let $\epsilon > 0$. Given a set of points P , denote by $\text{OPT}_1(P)$ the location of the optimal center
 384 of P (namely, the centroid of P in the case of the k -means problem or the median of P in
 385 the case of the k -median problem). We will make use of the following lemma of [21].

386 **► Lemma 21** (Lemma 5.3 in [21]). *Let P be a set of points in \mathbb{R}^d and X be a random sample
 387 of size $O(\epsilon^{-3} \log(1/\epsilon))$ from P and a and b such that $a \leq \text{cost}(P, \text{OPT}_1(P)) \leq b$. Then, we
 388 can construct a set Y of $O(2^{1/\epsilon^{O(1)}} \log(b/\epsilon a))$ points such that with constant probability there
 389 is at least one point $z \in X \cup Y$ satisfying $\text{cost}(P, \{z\}) \leq (1 + 2\epsilon)\text{cost}(P, \text{OPT}_1(P))$. Further,
 390 the time taken to construct Y from X is $O(2^{1/\epsilon^{O(1)}} \log(b/\epsilon a)d)$.*

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

- 392 1. Compute a coresets C for capacitated k -median as described by Lemma 21, and an estimate
 393 γ of the value of OPT using the classic $O(\log n)$ -approximation.
 394 In the remaining, we assume that the minimum pairwise distance between pairs of points
 395 of C is at least $\epsilon\gamma/(n \log n)$ since otherwise one can simply take a net of the input and
 396 the additive error is at most ϵOPT (see e.g.: [12]). Moreover, we assume that there is no
 397 cluster containing only one point of the coresets since these clusters can be “guessed” and
 398 dealt with separately.
- 399 2. Start with $\mathcal{C} = \emptyset$, then for each subset S of C of size $O(\epsilon^{-3} \log(k/\epsilon))$, for each $s = (1 + \epsilon)^i$
 400 in the interval $[\epsilon\gamma/(n \log n), \gamma]$ apply the procedure of Lemma 21 with $a = s$ and $b =$
 401 $(1 + \epsilon)a$ and add the output of the procedure to \mathcal{C} . We refer to \mathcal{C} as a set of approximate
 402 candidate centers.
- 403 3. Consider all subsets of size k of \mathcal{C} . For each subset, compute the cost of using this set
 404 of centers for the capacitated k -median instance by using a min cost flow computation.
 405 Output the set of centers of minimum cost.

406 We first discuss the running time of the algorithm. The time for computing the coresets
 407 is polynomial by Theorem 11. Generating \mathcal{C} takes $|\mathcal{C}|^{O(\epsilon^{-3} \log(1/\epsilon))} \cdot 2^{1/\epsilon^{O(1)}} \log((1 + \epsilon)/\epsilon)^d$
 408 time. For the last part, namely enumerating all subsets of \mathcal{C} of size k , the running time is
 409 $|\mathcal{C}|^{O(k\epsilon^{-3} \log(1/\epsilon))} \cdot 2^{k/\epsilon^{O(1)}} \log^k((1 + \epsilon)/\epsilon)$. Theorem 11 implies that $|\mathcal{C}| = \text{poly}(k \log n \epsilon^{-1})$ and
 410 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
 411 can be upper bounded by $(k/\epsilon)^{k\epsilon^{-O(1)}}$ or $n^{O(1)}$ based on whether or not $k\epsilon^{-O(1)} < \frac{\log n}{\log \log n}$,
 412 hence the improved running time in Theorem 3.

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

418 ► **Lemma 22.** *The \mathcal{C} computed by the algorithm contains a set of centers \tilde{S} that is such that*
 419 $\text{cost}(C, \tilde{S}) \leq (1 + \epsilon)\text{cost}(C, \text{OPT})$.

420 **Proof.** This follows almost immediately from Lemma 21. By Lemma 21, for each cluster C_i^*
 421 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
 422 procedure of Lemma 21 with the correct value of a to S_i^* yields a set of points containing a
 423 point z_i such that $\text{cost}(C_i^*, z_i) \leq (1 + 2\epsilon)\text{cost}(C_i^*, \text{OPT})$. Since the algorithm iterates over all
 424 subsets of size $O(\epsilon^{-3} \log(k/\epsilon))$, and that the pairwise distance is at least $\epsilon\text{OPT}/n$, it follows
 425 that S_i^* is one of the subset considered by the algorithm, and so z_i is part of \mathcal{C} . ◀

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

429 3.2 The Non-Uniform Case – Proof of Theorem 2

430 We now consider the non-uniform case. In this setting, the input consists of a set of points in
 431 \mathbb{R}^d together with a set of candidate centers in \mathbb{R}^d and a capacity η_f for each such candidate
 432 center. We make use of the following lemma. As slightly worse bound for the lemma can
 433 also be found in [26].

434 ► **Lemma 23** ([28]). *Let $\epsilon \in (0, 1)$ and $X \subseteq \mathbb{R}^d$ be arbitrary with X having size $n > 1$.*
 435 *There exists $f : \mathbb{R}^d \mapsto \mathbb{R}^m$ with $m = O(\epsilon^{-2} \log n)$ such that $\forall x \in X, \forall y \in \mathbb{R}^d, \|x - y\|_2 \leq$
 436 $\|f(x) - f(y)\|_2 \leq (1 + \epsilon)\|x - y\|_2$.*

23:12 On the Fixed-Parameter Tractability of Capacitated Clustering

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

- 439 1. Compute a coreset C for capacitated k -median as described by Theorem 21, and an
440 estimate γ of the value of OPT using the classic $O(\log n)$ -approximation.
441 In the remaining, we assume that the minimum pairwise distance between pairs of points
442 of C is at least $\epsilon\gamma/(n \log n)$ since otherwise one can simply take a net of the input and
443 the additive error is at most ϵOPT (see e.g.: [12]). Moreover, we assume that there is no
444 cluster containing only one point of the coreset since these clusters can be “guessed” and
445 dealt with separately.
- 446 2. Apply Lemma 23 to the points of the coreset to obtain a set of points in a Euclidean
447 space of dimension $\frac{\log k + \log \log n}{\epsilon^{O(1)}}$. Let C^* and A^* be respectively the image of the coreset
448 points and of the candidate centers through the projection.
- 449 3. Start with $\mathcal{V} = \emptyset$ For each point p of the coreset do the following: For each $i \in$
450 $\{1, 2, \dots, n^2\}$, consider the i th-ring defined by $\text{ball}(p, (1 + \epsilon)^i \epsilon\gamma/(n \log n)) \setminus \text{ball}(p, (1 +$
451 $\epsilon)^{i-1} \epsilon\gamma/(n \log n))$ and choose an $\epsilon \cdot (1 + \epsilon)^i \epsilon\gamma/(n \log n)$ -net. Consider the Voronoi diagram
452 induced by the points of the net. Then, for each Voronoi cell, add to \mathcal{V} the k candidate
453 centers of A^* in the cell that are of maximum capacity.
- 454 4. Enumerate all possible subset of \mathcal{V} of size k and output the one that leads to the solution
455 of minimum cost.

456 3.2.1 Correctness.

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

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

463 Therefore, it remains to show that the set \mathcal{V} contains a set of candidate centers that
464 yields a near-optimal solution. To see this, consider each center of the optimal solution in A^* .
465 For each such optimal center f , consider the closest coreset point $c(f)$ together with the ring
466 of $c(f)$ containing f . Let j be the index of this ring, namely $f \in \text{ball}(p, (1 + \epsilon)^j \epsilon\gamma/(n \log n)) \setminus$
467 $\text{ball}(p, (1 + \epsilon)^{j-1} \epsilon\gamma/(n \log n))$.

468 By definition of the net, there exists a point p of the net at distance at most $\epsilon \cdot \text{ball}(p, (1 +$
469 $\epsilon)^j \epsilon\gamma/(n \log n)) \leq 2\epsilon \|c - c(f)\|_2$ from $c(f)$. Therefore, consider the Voronoi cell of p and the
470 top- k candidate centers in terms of capacity. If f is part of this top- k , then f is part of \mathcal{V}
471 and we are done. Otherwise, it is possible to associate to f a center f^* that has capacity at
472 least the capacity of f , and so for all the optimal centers simultaneously since we consider
473 the top- k . Therefore, consider replacing f by f^* in the optimal solution. The change in cost
474 is at most, by the triangle inequality, $4\epsilon \|c - c(f)\|_2$ since both centers are in the Voronoi
475 cell of p . Finally, since c is the closest client to $c(f)$, the cost increases by a factor at most
476 $(1 + 4\epsilon)$ for each client and the correctness follows.

477 3.2.2 Running time.

478 We now bound the running time. The first two steps are clearly polynomial time. An
479 $\epsilon \cdot (1 + \epsilon)^i \epsilon\gamma/(n \log n)$ -net of a ball of radius $(1 + \epsilon)^i \epsilon\gamma/(n \log n)$ has size $\epsilon^{-O(d)}$ and so in this
480 context, after Step 2, a size $\epsilon^{-\left(\frac{\log k + \log \log n}{\epsilon^{O(1)}}\right)}$. Since for each element of the net, k centers are

481 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
 482 most $|C|k\epsilon^{-2} \log n \epsilon^{-\left(\frac{\log k + \log \log n}{\epsilon^{O(1)}}\right)}$ which is at most $|C|\epsilon^{-2}(k \log n)^{\epsilon^{-O(1)}} = (k\epsilon^{-1} \log n)^{\epsilon^{-O(1)}}$.
 483 Enumerating all subsets of size k takes time $(k\epsilon^{-1} \log n)^{k\epsilon^{-O(1)}}$ and the theorem follows.

 484 ——— **References** ———

- 485 **1** M. Adamczyk, J. Byrka, J. Marcinkowski, S. M. Meesum, and M. Włodarczyk. Constant
 486 factor FPT approximation for capacitated k-median. *ArXiv e-prints*, September 2018. arXiv:
 487 1809.05791.
- 488 **2** Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2016.
- 489 **3** Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for Euclidean k -
 490 medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on the*
 491 *Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 106–113, 1998. Available
 492 from: <http://doi.acm.org/10.1145/276698.276718>, doi:10.1145/276698.276718.
- 493 **4** Jarosław Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor
 494 approximation algorithms for hard capacitated k-median problems. In *Proceedings of the*
 495 *twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 722–736. SIAM,
 496 2014.
- 497 **5** Jarosław Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for
 498 uniform capacitated k-median problem with $1+\epsilon$ capacity violation. In *International Conference*
 499 *on Integer Programming and Combinatorial Optimization*, pages 262–274. Springer, 2016.
- 500 **6** Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees:
 501 Deterministic approximation algorithms for group steiner trees and k-median. In *STOC*,
 502 volume 98, pages 114–123. Citeseer, 1998.
- 503 **7** Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor ap-
 504 proximation algorithm for the k-median problem. *Journal of Computer and System Sciences*,
 505 65(1):129–149, 2002.
- 506 **8** K. Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces
 507 and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.
- 508 **9** Julia Chuzhoy and Yuval Rabani. Approximating k-median with non-uniform capacities. In
 509 *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '05*,
 510 pages 952–958, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
 511 Available from: <http://dl.acm.org/citation.cfm?id=1070432.1070569>.
- 512 **10** Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics.
 513 *CoRR*, abs/1812.07721, 2018. Available from: <http://arxiv.org/abs/1812.07721>, arXiv:
 514 1812.07721.
- 515 **11** Vincent Cohen-Addad, Arnaud de Mesmay, Eva Rotenberg, and Alan Roytman. The bane
 516 of low-dimensionality clustering. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM*
 517 *Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10,*
 518 *2018*, pages 441–456, 2018. Available from: <https://doi.org/10.1137/1.9781611975031.30>,
 519 doi:10.1137/1.9781611975031.30.
- 520 **12** Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight
 521 FPT approximations for k-median and k-means. *CoRR*, 2019.
- 522 **13** Wenceslas Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani.
 523 Approximation schemes for clustering problems. In Lawrence L. Larmore and Michel X.
 524 Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Com-*
 525 *puting, June 9-11, 2003, San Diego, CA, USA*, pages 50–58. ACM, 2003. Available from:
 526 <http://doi.acm.org/10.1145/780542.780550>, doi:10.1145/780542.780550.
- 527 **14** H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with
 528 $(1+\epsilon)$ -capacity violation. In *43rd International Colloquium on Automata, Languages,*

- 529 *and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016. Avail-
530 able from: <https://doi.org/10.4230/LIPIcs.ICALP.2016.73>, doi:10.4230/LIPIcs.ICALP.
531 2016.73.
- 532 **15** G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *STOC*, pages
533 209–217, 2005.
- 534 **16** Anupam Gupta, Euiwoong Lee, and Jason Li. An fpt algorithm beating 2-approximation
535 for k-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete*
536 *Algorithms, SODA '18*, pages 2821–2837, Philadelphia, PA, USA, 2018. Society for Industrial
537 and Applied Mathematics. Available from: [http://dl.acm.org/citation.cfm?id=3174304.](http://dl.acm.org/citation.cfm?id=3174304.3175483)
538 3175483.
- 539 **17** Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michal Włodarczyk. Losing
540 treewidth by separating subsets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium*
541 *on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages
542 1731–1749, 2019. Available from: <https://doi.org/10.1137/1.9781611975482.104>, doi:
543 10.1137/1.9781611975482.104.
- 544 **18** Sarel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering.
545 *Discrete & Computational Geometry*, 37(1):3–19, 2007. Available from: [http://dx.doi.org/](http://dx.doi.org/10.1007/s00454-006-1271-x)
546 [10.1007/s00454-006-1271-x](http://dx.doi.org/10.1007/s00454-006-1271-x), doi:10.1007/s00454-006-1271-x.
- 547 **19** Sarel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering.
548 In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL,*
549 *USA, June 13-16, 2004*, pages 291–300, 2004. Available from: [http://doi.acm.org/10.1145/](http://doi.acm.org/10.1145/1007352.1007400)
550 [1007352.1007400](http://doi.acm.org/10.1145/1007352.1007400), doi:10.1145/1007352.1007400.
- 551 **20** Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1 + \epsilon)$ -approximation
552 algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Annual IEEE*
553 *Symposium on Foundations of Computer Science, FOCS '04*, pages 454–462, Washington,
554 DC, USA, 2004. IEEE Computer Society. Available from: [http://dx.doi.org/10.1109/FOCS.](http://dx.doi.org/10.1109/FOCS.2004.7)
555 [2004.7](http://dx.doi.org/10.1109/FOCS.2004.7), doi:10.1109/FOCS.2004.7.
- 556 **21** Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes
557 for clustering problems in any dimensions. *J. ACM*, 57(2), 2010. Available from: <http://doi.acm.org/10.1145/1667053.1667054>, doi:10.1145/1667053.1667054.
- 558 **22** Euiwoong Lee. Partitioning a graph into small pieces with applications to path trans-
559 versal. *Mathematical Programming*, Mar 2018. Available from: [https://doi.org/10.1007/](https://doi.org/10.1007/s10107-018-1255-7)
560 [s10107-018-1255-7](https://doi.org/10.1007/s10107-018-1255-7), doi:10.1007/s10107-018-1255-7.
- 561 **23** Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings*
562 *of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015,*
563 *San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015. Available from: <https://doi.org/10.1137/1.9781611973730.47>, doi:10.1137/1.9781611973730.47.
- 564 **24** Shi Li. Approximating capacitated k-median with $(1 + k)$ open facilities. In *Proceedings of*
565 *the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016,*
566 *Arlington, VA, USA, January 10-12, 2016*, pages 786–796, 2016. Available from: <https://doi.org/10.1137/1.9781611974331.ch56>, doi:10.1137/1.9781611974331.ch56.
- 567 **25** Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. *ACM Trans.*
568 *Algorithms*, 13(2):22:1–22:18, 2017. Available from: <http://doi.acm.org/10.1145/2983633>,
569 doi:10.1145/2983633.
- 570 **26** Sepideh Mahabadi, Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Non-
571 linear dimension reduction via outer bi-lipschitz extensions. In *Proceedings of the 50th Annual*
572 *ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1088–1101, New York,
573 NY, USA, 2018. ACM. Available from: <http://doi.acm.org/10.1145/3188745.3188828>,
574 doi:10.1145/3188745.3188828.
- 575 **27** Dániel Marx and Michal Pilipczuk. Optimal parameterized algorithms for planar facil-
576 ity location problems using voronoi diagrams. In *Algorithms - ESA 2015 - 23rd An-*
577 *ual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages
578 579 580

- 581 865–877, 2015. Available from: https://doi.org/10.1007/978-3-662-48350-3_72, doi:
582 10.1007/978-3-662-48350-3_72.
- 583 **28** Shyam Narayanan and Jelani Nelson. Optimal terminal dimensionality reduction in euclidean
584 space. *CoRR – To appear in the proceedings of STOC’19*, abs/1810.09250, 2018. Available
585 from: <http://arxiv.org/abs/1810.09250>, arXiv:1810.09250.
- 586 **29** Yicheng Xu, Yong Zhang, and Yifei Zou. A constant parameterized approximation for hard-
587 capacitated k-means. *CoRR*, abs/1901.04628, 2019. Available from: [http://arxiv.org/abs/](http://arxiv.org/abs/1901.04628)
588 [1901.04628](http://arxiv.org/abs/1901.04628), arXiv:1901.04628.