On Tight FPT Time Approximation Algorithms for k-Clustering Problems

Han Dai *
School of Computer Science,
Nanjing University,
Nanjing, Jiangsu Province, China.
handai@smail.nju.edu.cn

Shi Li *
School of Computer Science,
Nanjing University,
Nanjing, Jiangsu Province, China.
shili@nju.edu.cn

Sijin Peng
CSAIL,
Massachusetts Institute of Technology,
Cambridge, MA, United States.
sijinp@mit.edu

Abstract

Following recent advances in combining approximation algorithms with fixed-parameter tractability (FPT), we study FPT-time approximation algorithms for minimum-norm k-clustering problems, parameterized by the number k of open facilities.

For the capacitated setting, we give a tight $(3+\epsilon)$-approximation for the general-norm capacitated k-clustering problem in FPT-time parameterized by k and ϵ. Prior to our work, such a result was only known for the capacitated k-median problem [29]. As a special case, our result yields an FPT-time 3-approximation for capacitated k-center. The problem has not been studied in the FPT-time setting, with the previous best known polynomial-time approximation ratio being 9 [6].

In the uncapacitated setting, we consider the `top`-cn norm k-clustering problem, where the goal of the problem is to minimize the `top`-cn norm of the connection distance vector. Our main result is a tight $\left(1+\frac{2}{ec}+\epsilon\right)$-approximation algorithm for the problem with $c \in \left(\frac{1}{e}, 1\right]$. (For the case $c \le \frac{1}{e}$, there is a simple tight $(3+\epsilon)$-approximation.) Our framework can be easily extended to give a tight $\left(3, 1+\frac{2}{e}+\epsilon\right)$-bicriteria approximation for the (k-center, k-median) problem in FPT time, improving the previous best polynomial-time $(4, 8)$ guarantee [5].

All results are based on a unified framework: computing a $(1+\epsilon)$-approximate solution using $O\left(\frac{k \log n}{\epsilon}\right)$ facilities S via LP rounding, sampling a few client representatives R based on the solution S, guessing a few pivots from $S \cup R$ and some radius information on the pivots, and solving the problem using the guesses. We believe this framework can lead to further results on k-clustering problems.

1 Introduction

In the supplier setting of clustering problems, we are given a set F of facilities, a set C of n clients, a metric d over $F \cup C$, and a non-negative integer k. The objective is to open a set $S \subseteq F$ of k facilities and assign each client to an open facility, so as to minimize some function on the connection distances. Chakrabarty and Swamy [17] introduced the *minimum-norm k-clustering problem*, where the goal is to minimize a monotone symmetric norm $f : \mathbb{R}^C_{\ge 0} \to \mathbb{R}_{\ge 0}$ applied to the connection distances. Specifically, the objective is to find a set $S \subseteq F$ of k facilities so as to minimize $f((d(j,S))_{j\in C})$, where $d(j,S) := \min_{i\in S} d(j,i)$.

*The work of HD and SL is supported by State Key Laboratory for Novel Software Technology, and New Cornerstone Science Foundation.

When the norm f is L_∞, L_1 or L_2 norm, the problem respectively corresponds to the well-known *k-supplier-center*, *k-median* and *metric k-means*[1] problems. These problems have been extensively studied in the literature [4, 19, 14, 44, 39, 25, 34, 32, 47, 24, 35, 38, 51]. The best known approximation ratios for the three problems are respectively 3 [58], $2+\epsilon$ [26] and 6.357 [4], while the known hardness of approximation bounds are $3, 1+\frac{2}{e}$ and $1+\frac{8}{e}$ [45, 42].

Two other important classes of monotone symmetric norms studied in the literature are the $\mathtt{top}\text{-}\ell$ norms and *ordered* norms. For any $\ell \in [n]$, the $\mathtt{top}\text{-}\ell$ norm of $v \in \mathbb{R}^n_{\geq 0}$ is the sum of ℓ largest coordinates of v. An ordered norm is a non-negative linear combination of several $\mathtt{top}\text{-}\ell$ norms. The clustering problem with this objective is called the *ordered k-median* problem. Independently, Byrka, Sornat, and Spoerhase [16] and Chakrabarty and Swamy [17] gave the first constant-factor approximation algorithms for ordered k-median. The approximation ratio was subsequently improved by Chakrabarty and Swamy [18] to $5+\epsilon$.

Another aspect that naturally arises in clustering problems is the presence of facility capacities. In the supplier setting, a facility may not be able to serve too many clients. In clustering applications, there may be upper bounds on the sizes of clusters. In the *minimum-norm capacitated k-clustering* problem, each facility $i \in F$ is additionally given a capacity $u_i \in \mathbb{Z}_{>0}$. In a feasible solution, every facility $i \in S$ can serve at most u_i clients. Formally, we need to output a set $S \subseteq F$ of k facilities and an assignment vector $\sigma \in S^C$ such that $|\sigma^{-1}(i)| \leq u_i$ for every $i \in S$, so as to minimize $f(d(j, \sigma_j)_{j \in C})$.

The main problems that have been studied in the capacitated setting are the capacitated k-center and capacitated k-median problems. For capacitated k-center, the current best approximation ratio is 9, due to [6], which improves upon the earlier $O(1)$-approximation of Cygan, Hajiaghayi and Khuller [30]. When capacities are uniform, a better approximation ratio of 6 is known [9, 50]. In contrast, the approximation status of the capacitated k-median problem is less satisfactory. Many $O(1)$ bicriteria approximation algorithms are known, which either violate the limit k on the number of open facilities [1] or the capacity constraints [23, 52, 12]. In both settings, the violation factor can be reduced to $1+\epsilon$ [54, 53, 15, 31]. For true approximation algorithms, the folklore result of $O(\log n)$-distortion embedding of any n-point metric into a distribution of HST metrics [33] leads to an $O(\log n)$-approximation for the problem. This ratio was improved to $O(\log k)$ by Adamczyk, Byrka, Marcinkowski, Meesum and Wlodarczyk [3], who applied the HST embedding technique, but on a k-point metric obtained by considering the k-center objective. This remains the the current best approximation ratio for the capacitated k-median problem, and obtaining a true $O(1)$-approximation for this problem—without any violation—remains a notorious open question.

Recently, combining paradigms of approximation algorithms and *Fixed-Parameter-Tractability (FPT)* has led to exciting new results, especially for problems where the improvement on polynomial time approximation algorithms has stalled, including many clustering problems. In many clustering applications, k is a small number, allowing us to have a running time of the form $g(k) \cdot n^{O(1)}$, where g is a function only on k. Such a running time is called FPT time, parameterized by k.

Cohen-Addad, Gupta, Kumar, Lee and Li [28] studied the k-median and metric k-means problems in FPT time, obtaining approximation factors of $1+\frac{2}{e}+\epsilon$ and $1+\frac{8}{e}+\epsilon$ respectively. On the negative side, these approximation factors are tight in FPT time under the assumption FPT $\neq W[1]$ [28]. For the capacitated k-median problem, Cohen-Addad and Li [29] developed a tight 3-approximation in FPT time. More recently, [2] proposed an Efficient Parameterized Approximation Scheme (EPAS) for the minimum-norm k-clustering problem on metrics with bounded ϵ-scatter dimension, which include Euclidean metrics, metrics of bounded doubling dimension and planar metrics. This scheme achieves a $(1+\epsilon)$-approximation in $g(k,\epsilon) \cdot \mathrm{poly}(n)$ time, for some function g on k and ϵ.

Goyal and Jaiswal [40] studied FPT time approximation algorithms for a family of constrained clustering problems, with the objective of minimizing the z-th power of the maximum connection distance for any $z > 0$. A main constraint they consider is the *cluster-size constraint*: we are given k size bounds $r_1, r_2, \cdots, r_k \geq 0$, and the partition $(O_1, O_2, \cdots, O_k)$ of clients must satisfy $|O_1| \leq r_1, |O_2| \leq r_2, \cdots, |O_k| \leq r_k$ (the partition can be arbitrarily permuted). They also considered many other constraints such as lower bounds on cluster sizes, color constraints, and fault-tolerant requirements. They developed a general FPT-time framework for

[1]The objective of the metric k-means problem is to minimize the sum of squared distances, which is equivalent to minimize the L_2 norm.

these problems that achieve an approximation ratio of 2^z for k-center-type problems (where facilities can be put anywhere in the metric), and 3^z for k-supplier-type problems (where facilities can only be put at specified locations). Both ratios were shown to be tight under GAP-ETH.

In particular, the problem most closely related to the capacitated k-center problem we study is the r-capacity k-supplier problem. The goal is to find a clustering of clients satisfying the cluster-size constraint and choose a facility for each cluster, so as to minimize the maximum connection distance. When all size bounds $r_1, r_2, \cdots, r_k$ are the same, the problem reduces to the soft and uniform capacitated k-center problem where each facility i has capacity $u_i = r_1$. However, as noted in [40], for general (non-uniform) size bounds, there is no easy reduction between the two problems, since one imposes capacities on clusters whereas the other imposes capacities on facilities.

1.1 Our Results

Following this line of research, we study approximation algorithms on k-clustering problems in FPT time, with k being the parameter.

Our first result is a $(3+\epsilon)$-approximation algorithm for the minimum-norm capacitated k-clustering problem in FPT time parameterized by k and ϵ.

Theorem 1.1. *For any $\epsilon > 0$, there is a $(3+\epsilon)$-approximation algorithm for the minimum-norm capacitated k-clustering problem, that runs in time $g(k, \epsilon) \cdot \mathrm{poly}(n)$, where g is a computable function depending on k and ϵ.*

In particular, this implies

Corollary 1.2. *There is a 3-approximation algorithm for the capacitated k-center problem, that runs in time $g(k) \cdot \mathrm{poly}(n)$, where g is a computable function depending on k.* [2]

To the best of our knowledge, the result was previously only known for capacitated k-median [29]. In particular, it was not known even for the capacitated k-center problem with general capacities. The best known approximation ratio for the problem remains 9, achieved by a polynomial time algorithm due to [6]. Our result gives a 3-approximation in FPT time. This is tight under the assumption that FPT $\neq W[1]$, even for the (uncapacitated) k-supplier center problem.

We then turn to the uncapacitated setting. Our main focus is the $\texttt{top-}cn$ norm for a constant $c \in (0, 1]$. (The problem can be refered to as the cn-centrum problem in the literature.) On the negative side, the problem is hard to approximate within a factor better than $\min\{3, 1+\frac{2}{ec}\}$ in FPT time, assuming FPT $\neq W[1]$ [57]. We complement this with a matching positive result:

Theorem 1.3. *For any $\epsilon > 0$, there is a $\left(\min\left\{3, 1+\frac{2}{ec}\right\}+\epsilon\right)$-approximation algorithm for the $\texttt{top-}cn$ norm k-clustering problem with running time $g(k, \epsilon) \cdot n^{O(1)}$, where g is a computable function on k and ϵ.*

The theorem suggests that considering the $\texttt{top-}\ell$ norm for $\ell = cn$ is appropriate, as the approximation ratio is a function of $\frac{\ell}{n}$. The negative result implies that if $\frac{\ell}{n} < \frac{1}{e}+\epsilon$, then the problem is hard to approximate within a factor better than 3 even in FPT time, assuming FPT $\neq W[1]$. On the other hand, achieving a $(3+\epsilon)$-approximation for minimum-norm k-clustering in FPT time is easy (See Corollary 2.10).

Finally, we show that the framework can be easily extended to give a $(3, 1+\frac{2}{e}+\epsilon)$-bi-criteria approximation for the clustering problem with both k-center and k-median objectives. This was introduced by Alamdari and Shmoys [5], who gave a polynomial-time $(4, 8)$-approximation. We improve the bi-criteria approximation factors to $(3, 1+\frac{2}{e}+\epsilon)$, albeit with FPT time. Indeed, our result works for the more general $\texttt{top-}cn$ norm. See Section E for formal definitions used in the theorem:

[2] For the k-center objective, $3+\epsilon$ can be reduced to 3. By guessing and scaling distances, we can assume the optimum value is 1. Changing the distance $d(i,j)$ to $\lceil d(i,j) \rceil$ does not change the optimum solution of the instance. For such an instance, a $(3+\epsilon)$-approximation is a 3-approximation if $\epsilon < 1$.

Theorem 1.4. *For a constant $\epsilon > 0$, there is a $\left(3, \min\left\{3, 1+\frac{2}{e}\right\}+\epsilon\right)$-bi-criteria approximation algorithm for the $(L_\infty, \texttt{top-}cn)$-norms k-clustering problem, with running time $g(k,\epsilon)\cdot \mathrm{poly}(n)$, for some computable function g on k and ϵ.*

1.2 Our Techniques

Overview of FPT Time $(3+\epsilon)$-Approximation for Minimum-Norm Capacitated k-Clustering In the overview, we mainly focus on the $\texttt{top-}\ell$ norm capacitated k-clustering problem, as it already captures the essence of our algorithm. The problem generalizes both capacitated k-center (with $\ell = 1$) and capacitated k-median (with $\ell = n$). Letting ℓ go from 1 to n gives a smooth transition from the k-center to the k-median objective. Therefore, we need to achieve $(3+\epsilon)$-approximation algorithms for both extreme cases while unifying their key ideas to handle the general ℓ case.

Our FPT time 3-approximation for capacitated k-center is new, and we sketch the intuition. For simplicity, we first focus on the soft-capacitated setting, where facilities may be opened multiple times. (The total number of open facilities, counting multiplicities, remains k.)

By guessing the optimum value, scaling and rounding distances, we can assume all distances are integers and the optimum value is 1. We construct a solution S with cost 1 and $O(k \log n)$ open facilities that respect the capacity constraints; this can be easiliy obtained via LP rounding. For each facility in S, we sample $O(\log n)$ clients connected to it. =

For an optimum cluster (i^*, J), i^* is the open facility and J is the assigned clients. If every $j \in J$ is connected to a facility i with $u_i \le |J|$ in the solution S, then with high probability, some client in J will be sampled. This sampled client j can then serve as a *pivot* for the cluster: we open the facility with the largest capacity within distance at most 1 of j.

On the other hand, if some $j \in J$ is connected to a facility $i \in S$ with capacity $u_i > |J|$, then we can directly use i as a replacement for i^*. In either case, by guessing, we can identify a facility i with $u_i \ge |J|$ and $d(i, i^*) \le 2$, which leads to a 3-approximation.

The hard-capacitated case can be addressed using the coloring idea [29], along with additional care. By guessing a coloring function, we can ensure with reasonble probability that the optimum facilities have distinct colors, which can be used to guide our selection of facilities. We have a stricter condition for the second case above: We require that $i \in S$ has capacity $u_i \ge k|J|$ so that i have enough capacity to serve k clusters in the optimum solution. We also need to take care of the case where an optimum facility also appears in S.

Our algorithm for the $\texttt{top-}\ell$ norm needs to capture an FPT time $(3+\epsilon)$-approximation for capacitated k-median. Such an algorithm was known [29]. However, it relies on the coreset technique, which has been extensively studied in the literature [11, 10, 46, 21, 22, 36]. However, the technique is based on sampling and only works well when $\ell = \Omega(n)$. This makes the algorithm hard to combine with the k-center objective. Instead, we use an alternative $(3+\epsilon)$-approximation for capacitated k-median that avoids coresets.

We obtain a $(1+\epsilon)$-approximation for the problem using $O(\frac{k \log n}{\epsilon})$ facilities S. We sample a few clients from each cluster in solution S. Additionally, we sample a few clients in C, with probabilities proportional to their costs in S. If for an optimum cluster (i^*, J), the total cost of J in the solution S is large, we likely sampled a "good" client in J, which can serve as a pivot. Otherwise, the total cost of J in the solution S can be essentially ignored, and some facility with large enough capacity in S can be used to replace i. Again, extra-care is needed to handle hard capacities.

To unify the approaches for both the capacitated k-center and k-median problems, we guess the ℓ-th largest connection distance t in the optimum solution. The $\texttt{top-}\ell$ norm of the optimum distance vector $v \in \mathbb{R}^C_{\ge 0}$ is $\ell t + \sum_{j \in C}(v_j - t)^+$. At a very high level, we treat the ℓt term as the k-center component, and the $\sum_{j\in C}(v_j - t)^+$ term as the k-median component. Our algorithm combines the techniques to handle these two parts effectively.

To extend the idea of a general monotone symmetric norm f, we apply the above idea for all distances t that is an integer power of $1+\epsilon$. This will bound the top-ℓ norm cost of the solution, for every $\ell \in [0, n]$. Every symmetric norm is the maximum of many *order norms*, which in turn are non-negative linear combinations of top-ℓ norms [17]. Therefore, a bound on the top-ℓ norms leads to a bound on the f-norm. This proves Theorem 1.1.

Overview of $\left(1+\frac{2}{ce}+\epsilon\right)$-Approximation for Top-$cn$ Norm (Uncapacitated) k-Clustering Problem for $c \in (\frac{1}{e}, 1]$ To describe our algorithm, we first give an overview of the $(1+\frac{2}{e})$-approximation for k-median in FPT time by [28]. By using the coreset technique, one can assume that the set of clients is located in only $O(\frac{k \log n}{\epsilon})$ positions. Given a cluster (i^*, J) in the optimum clustering $\mathcal{C}$, a *leader* of the cluster is defined as the client in J that is nearest to i^*.

The algorithm proceeds by guessing the k leaders and the approximate distances of the leaders to their respective centers. With this information, a $(1+\frac{2}{e})$-approximation can be obtained via submodular maximization. Roughly speaking, we consider a ball centered at each leader, with the radius equal to the guessed distance to its center. Ensuring that every ball contains one facility yields a cost of $3 \cdot \mathrm{opt}$. Choosing the optimum facilities within these balls improves the cost by $2 \cdot \mathrm{opt}$. The goal then becomes maximizing this improvement, which can be cast as a submodular maximization problem under a partition matroid constraint. The standard $(1-\frac{1}{e})$ approximation ratio for submodular maximization translates into an approximation ratio of $3 - 2(1-\frac{1}{e}) = 1 + \frac{2}{e}$ for k-median.

To generalize this approach to the `top`-cn setting, we modify the algorithm in two ways. First, one would need to construct a coreset for the `top`-cn objective. We believe this is possible for any constant $c \in (0,1)$; however, we instead adopt an alternative approach that is similar to our algorithm in the capacitated setting and avoids the use of coresets. We obtain a bi-criteria $(1+\epsilon)$-approximation for the problem by opening $O\left(\frac{k \log n}{\epsilon}\right)$ facilities S. We sample a few clients R with probability proportional to their costs in S, and then guess a set of k pivots from $R \cup S$. Pivots play a role similar to leaders but can be any points in the metric space, not necessarily the clients in the respective optimum clusters. With reasonable probability, when the guesses are correct, the pivots have the desired property that leaders have.

The second modification is essential. With the `top`-cn cost, the improvement function is no longer submodular. Instead, we use an LP-based approach to construct the set of centers. A useful notion here is the *occurrence time vector* of a cost vector: in such a vector δ, δ_t represents the number of times distance t occurs in the cost vector. We naturally allow occurrence times to be fractional. We then formulate an LP relaxation that uses the pivot and radius information, with the objective of minimizing the `top`-cn cost of the occurrence time vector δ.

We design a randomized rounding algorithm such that, in expectation, the occurrence time vector of the integral solution is no worse than $(1-\frac{1}{e})\delta + \frac{1}{e}(\delta \otimes 3)$, where $\delta \otimes 3$ denotes the vector obtained from δ by scaling all distances by a factor of 3. We bound the ratio of the `top`-cn cost of vector to that of δ. The worst-case scenario arises when the cost vector is the all-one vector, i.e., when the occurrence time vector has distance 1 appearing n times and all other distances appearing zero times. This yields a bound of $1 + \frac{2}{ec}$. Finally, using the occurrence time representation, the `top`-cn function is concave, leading to an overall approximation ratio of $1 + \frac{2}{ec} + \epsilon$. This proves Theorem 1.3.

A simple modification to the algorithm can handle the L_∞ norm, and simultaneously provide a 3-approximation for it, proving Theorem 1.4.

1.3 Other Related Work

For the k-supplier-center problem, the special case where $F = C$ has been studied more in the literature and is simply known as the k-center problem. Simple greedy algorithms achieve 2 and 3-approximations for the k-center and k-supplier-center problems respectively [44, 58]. They are the best possible for the two problems assuming P $\neq$ NP.

Charikar, Guha, Tardos, and Shmoys [20] gave the first constant factor approximation algorithms for the k-median problem, which is a $6\frac{2}{3}$-approximation based on LP rounding. Since then, there has been a long line of work aiming at improving the approximation ratio [7, 13, 19, 27]. In a recent milestone result, Cohen-Addad, Grandoni, Lee, Schwiegelshohn and Svensson [26], using the Lagrangian Multiplier Preserving (LMP) 2-approximation for the uncapacitated facility location problem due to Jain and Vazirani

[48], together with a novel framework for converting this algorithm into one for k-median with only a loss of $1+\epsilon$ in the approximation ratio.

For the metric k-means problem, Gupta and Tangwongsan [43] provided the first $O(1)$-approximation, which is a 16-approximation based on local search. This factor was later improved by [4] to $9+\epsilon$. A more commonly used variant in unsupervised machine learning, which is simply referred to as the k-means problem, considers the case where $C \subseteq F = \mathbb{R}^\ell$ and d is the Euclidean metric. A widely used heuristic for the problem is Lloyd's algorithm [56]. For k-means, [49] gave a $(9+\epsilon)$-approximation algorithm, which was improved to 6.357 by [4].

For the capacitated k-median problem, there is a line of work that provides polynomial time bi-criteria approximation algorithms, which either violate the number k of open facilities [1], or the capacity constraints [23, 52, 12]. These algorithms are based on the natural LP relaxation for the problem. There are gap instances showing that, in order to obtain an $O(1)$-approximation, the violation factors for the number of open facilities and the capacities constraints must be at least 2. This barrier was overcome by Li [54], who gave an $O(1)$-approximation for the problem with uniform capacities by opening $(1+\epsilon)k$ facilities, using a stronger LP relaxation. This result has since been generalized to the non-uniform setting [53]. For $O(1)$-approximations with capacity violations, the violation factor has also been reduced to $(1+\epsilon)$ in both the uniform-capacitated [15] and non-uniform capacitated [31] versions of the problem.

Adamczyk, Byrka, Marcinkowski, Meesum and Wlodarczyk [3] gave a $(7+\epsilon)$-approximation for the capacitated k-median problem in FPT time, which is the first $O(1)$-approximation for the problem in the FPT setting. This approximation factor was later improved to $3+\epsilon$ by [29], who also provided a $(1+\epsilon)$-approximation for the problem in Euclidean spaces of arbitrary dimensions. Both algorithms run in FPT time. [59] presented an FPT-time $(69+\epsilon)$-approximation algorithm for the hard-capacitated k-means problem in Euclidean spaces. Some other variants of k-clustering problems in the FPT-time region were also studied [41, 37, 8]. Recently, Liu, Chen and Xu [55] studied the uniform capacitated k-supplier problem in FPT time and obtained a 5-approximation algorithm.

Organization In Section 2, we give some preliminaries. In Section 3, we give the $(3+\epsilon)$-approximation for the minimum-norm capacitated k-clustering problem in FPT time, proving Theorem 1.1. In Section 4, we give the tight $\left(\min\left\{1+\frac{2}{ec}, 3\right\}+\epsilon\right)$-approximation algorithm for the $\texttt{top-}cn$ norm k-clustering problem, proving Theorem 1.3. All missing proofs can be found in the appendix; in particular, we defer the proof of Theorem 1.4 to Appendix E.

2 Preliminaries

For any $a \in \mathbb{R}$, we let $(a)^+ = \max\{a, 0\}$. For every real-valued vector y over some domain U and any $S \subseteq U$, we let $y(S) := \sum_{i\in S} y_i$, unless otherwise defined.

Given a metric space (V, d), $v \in V$ and $S \subseteq V$, we define $d(v, S) := \min_{i\in S} d(v, i)$ to be the distance from v to the set S. For any subset $U \subseteq V, v \in V$ and $r \geq 0$, we define $\texttt{ball}_U(v, r) := \{u \in U : d(u, v) \leq r\}$ to be the set of points in U with distance at most r to v.

2.1 Norms

Definition 2.1 (Norms). *A function $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a norm, if it satisfies the following 3 properties:*

(2.1a) $f(x) = 0$ iff $x = 0$ (non-negativity).

(2.1b) For any $x \in \mathbb{R}^n_{\geq 0}, \lambda \in \mathbb{R}_{\geq 0}$, we have $f(\lambda x) = \lambda f(x)$ (homogeneity).

(2.1c) For any $x, y \in \mathbb{R}^n_{\geq 0}$, we have $f(x + y) \leq f(x) + f(y)$ (subadditivity).

For a norm f, we shall also use $|x|_f$ to denote $f(x)$, the f-norm of the vector x.

Definition 2.2 (Monotone and Symmetric Norms)**.** *A norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be monotone, if for any $x, y \in \mathbb{R}^n_{\geq 0}$ satisfying $x \leq y$, we have $f(x) \leq f(y)$. It is said to be symmetric, if for any $x \in \mathbb{R}^n_{\geq 0}$, and any permutation matrix $A \in \{0,1\}^{n \times n}$, we have $f(x) = f(Ax)$.*

Here are some common monotone symmetric norms studied in the literature:

- L_p norms, $p \geq 1$: $L_p(x) := \left(\sum_{j=1}^n x_j^p\right)^{1/p}, \forall x \in \mathbb{R}^n_{\geq 0}$. We also use $|x|_p$ for $L_p(x)$.
- L_∞ norm: $L_\infty(x) := \max_{j \in [n]} x_j, \forall x \in \mathbb{R}^n_{\geq 0}$. We also use $|x|_\infty$ for $L_\infty(x)$.
- Top-ℓ norm, $\ell \in [0, n]$ is a real: $|x|_{\texttt{top-}\ell}$ for any $x \in \mathbb{R}^n_{\geq 0}$ is the maximum of $\alpha^{\mathrm{T}} x$ over all $\alpha \in [0,1]^n$ satisfying $|\alpha|_1 = \ell$. When ℓ is an integer, $x_{\texttt{top-}\ell}$ is the sum of the ℓ largest coordinates in x. In this paper, we shall allow ℓ to take fractional values. Notice that $\texttt{top-}1 \equiv L_\infty$ and $\texttt{top-}n \equiv L_1$.
- Ordered norms: This is a non-negative linear combination of top-ℓ norms, for different values of ℓ. See the following definition.

Definition 2.3 (Ordered norms)**.** *Given a non-negative and non-increasing weight vector $w = (w_1, \ldots, w_n)$. The w-ordered norm of any vector $x \in \mathbb{R}^n_{\geq 0}$, denoted as $|x|_{w\texttt{-ordered}}$, is defined as follows:*

$$|x|_{w\texttt{-ordered}} := \sum_{i=1}^n w_i x_i^{\downarrow},$$

where $x_i^{\downarrow}$ is the i-th largest value in $\{x_1, x_2, \cdots, x_n\}$ (counting multiplicities).

The following folklore lemma is useful when analyzing top-ℓ norms:

Lemma 2.4. *For any $x \in \mathbb{R}^n_{\geq 0}$, we have $|x|_{\texttt{top-}\ell} = \min_{t \geq 0}\left(\sum_{j=1}^n (x_j - t)^+ + \ell t\right)$. Moreover, the minimum is achieved when t is the ℓ-th largest coordinate in x.*

The following observation is easy to see:

Observation 2.5. *Given a non-negative and non-increasing weight vector $w = (w_1, \ldots, w_n)$. Let $w_{n+1} = 0$. Then for any $x \in \mathbb{R}^n_{\geq 0}$, we have $|x|_{w\texttt{-ordered}} = \sum_{\ell=1}^n (w_\ell - w_{\ell+1}) \cdot |x|_{\texttt{top-}\ell}$.*

Any monotone symmetric norm can be expressed as the maximum of ordered norms [17]:

Lemma 2.6 ([17])**.** *Let $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a monotone symmetric norm. There exists a closed set $\mathcal{W}$ of non-increasing vectors in $\mathbb{R}^n_{\geq 0}$, such that for any $x \in \mathbb{R}^n_{\geq 0}$, we have*

$$|x|_f = \max_{w \in \mathcal{W}} |x|_{w\texttt{-ordered}} = \max_{w \in \mathcal{W}} \sum_{\ell=1}^n (w_\ell - w_{\ell+1}) |x|_{\texttt{top-}\ell}.$$

2.2 Definitions of Problems

Definition 2.7 (Minimum-Norm k-Clustering)**.** *Let $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a monotone symmetric norm. In the Minimum-Norm k-Clustering problem under f-norm, we are given a set F of facilities, a set C of n clients, a positive integer $k \leq |F|$, and a metric space $(F \uplus C, d)$ over $F \uplus C$. The goal of the problem is to find a set $S \subseteq F$ with $|S| = k$ so as to minimize $f\big((d(j,S))_{j \in C}\big)$.*

When f are the L_∞, L_1 and L_2 norms, the problem becomes k-supplier-center, k-median and metric k-means problems respectively.

Definition 2.8 (Minimum-Norm Capacitated k-Clustering)**.** *Let $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a monotone symmetric norm. In the Minimum-Norm Capacitated k-Clustering problem under f-norm, we are given F, C, n, k and d as in Definition 2.7. Additionally, we are given a capacity $u_i \in \mathbb{Z}_{\geq 0}$ for every $i \in F$. The goal of the problem is to find a set $S \subseteq F$ with $|S| = k$, and an assignment vector $\sigma \in S^C$ such that $|\sigma^{-1}(i)| \leq u_i$ for every $i \in S$, so as to minimize $f\big((d(j, \sigma_j))_{j \in C}\big)$.*

When f are the L_∞ and L_1 norms, the problem becomes capacitated k-center[3] and capacitated k-median respectively.

We can assume distances are either ∞ or integers in $[0, \mathrm{poly}(n)]$, by losing a factor of $1 + \epsilon$ in the approximation ratio. See Appendix A. As is typical, we do sampling and make guesses during our algorithms. When analyzing the algorithm, we define the conditions for the sampling steps being successful, and the correct answers for all the guesses. In the actual algorithm, we have to enumerate all possibilities for the guesses. It suffices for our algorithm to succeed with $\frac{1}{g(k,\epsilon)\mathrm{poly}(n)}$ probability, as we can repeat it many times and output the best solution generated, to increase the probability to $1 - 1/\mathrm{poly}(n)$. For the guessing and repetition to be feasible, it is required that we can compute the cost of the constructed solutions. This is typically the case.

2.3 $(1+\epsilon)$-Approximation for Capacitated k-Clustering with $O(\frac{k \log n}{\epsilon})$ Facilities

It is useful to run an LP rounding algorithm to obtain a $(1+\epsilon)$-approximate solution for capacitated k-clustering with $O(\frac{k \log n}{\epsilon})$ facilities S. With FPT time allowed, we can afford to guess $\mathrm{poly}(k, \frac{1}{\epsilon})$ facilities in S.

We are given F, C, n, d, k and capacities $(u_i)_{i \in F}$ as in a capacitated k-clustering problem. Instead of a monotone symmetric norm f, we are given a convex monotone function $h : \mathbb{R}^C_{\geq 0} \to \mathbb{R}_{\geq 0}$ (which is not necessarily symmetric or a norm). The goal is the same as that of the minimum-norm capacitated k-clustering problem, except that we are minimizing the h function of the connection distance vector. Let opt' be the value of the instance.

For some $i \in F$ and $J \subseteq C$, we say (i, J) is valid star if $|J| \leq u_i$. The following theorem summarizes the $(1+\epsilon)$-approximation, whose proof is deferred to Appendix B.

Theorem 2.9. *Let $\epsilon > 0$ be a constant. We can efficiently find a set $\mathcal{S}$ of $O\left(\frac{k \ln n}{\epsilon}\right)$ valid stars such that*

- $\biguplus_{(i,J) \in \mathcal{S}} J = C$.
- *For every $j \in C$, let $b = d(i, j)$ for the unique star (i, J) with $j \in J$. Then $h\left((1-\epsilon)b\right) \leq \mathrm{opt}'$.*

The algorithm succeeds with high probability.

The following corollary is an immediate consequence of our analysis, and we were unable to find an explicit statement of the result in the literature. For the `top`-cn norm k-clustering problem, we need to use it to handle the case $c \leq \frac{1}{e}$. We defer the Proof to Appendix B.

Corollary 2.10. *Consider the minimum-norm k-clustering problem under a monotone norm f (which is not necessarily symmetric). There is a $(3+\epsilon)$-approximation algorithm for the problem in FPT time with parameters k and ϵ.*

2.4 Finding $(1+\epsilon)$-Approximate Assignment for Minimum-Norm Capacitated k-Clustering with Open Facilities

Consider an instance of the Minimum-Norm Capacitated k-Clustering problem. For many natural norms such as L_p and `top`-ℓ norms, when given the set S of k open facilities, the optimum assignment of C to S can

[3] In the literature, capacitated k-center refers to the problem where F and C can be unrelated, instead of the problem where $F = C$.

be found easily. However, this is not trivial for general monontone symmetric norms f. Instead, we present an FPT-time algorithm to find a $(1+\epsilon)$-approximate assignment. The proof of the following theorem is described in Appendix C.

Theorem 2.11. *Consider a minimum-norm capacitated k-clustering instance defined by F, C, n, d, u and k, under a symmetric monotone norm f. Assume $|F| = n = k$ and $\epsilon > 0$ is a constant. We can find a $(1+\epsilon)$-approximate assignment $\sigma \in F^C$ for the instance in time $g(k,\epsilon) \cdot \mathrm{poly}(n)$, for a computable function g depending on k and ϵ.*

3 FPT Time $(3+\epsilon)$-Approximation for Minimum-Norm Capacitated k-Clustering

In this section, we give the FPT time $(3+\epsilon)$-approximation algorithm for the minimum-norm capacitated k-clustering problem, proving Theorem 1.1. Recall that we say $(i \in F, J \subseteq C)$ is a valid star if $|J| \le u_i$. We fix an optimum solution $\mathcal{S}^*$ of k valid stars that is unknown to the algorithm. $\mathcal{S}^*$ cover all clients in C, and all facilities are distinct. Let S^* be the set of facilities in $\mathcal{S}^*$. So, we have $|S^*| = |\mathcal{S}^*| = k$. Let opt be the f-norm cost of $\mathcal{S}^*$.

Algorithm 1 $(3+\epsilon)$-Approximation for Minimum-Norm Capacitated k-Clustering

1: define set $\mathbf{T}$ of non-negative reals with $|\mathbf{T}| \le O\left(\frac{\log n}{\epsilon}\right)$, as described in text
2: **for** every $t \in \mathbf{T}$ **do**
3: with objective $h(v) := \sum_{j \in C}(v_j - t)^+$ for every $v \in \mathbb{R}^C_{\ge 0}$, use Theorem 2.9 to obtain a set $\mathcal{S}^t$ of $O\left(\frac{k \ln n}{\epsilon}\right)$ valid stars
4: $S^t \leftarrow \{i \in F : \exists J, (i,J) \in \mathcal{S}^t\}$
5: randomly choose a color function $\texttt{color} : F \to [k]$
6: guess the types of each color $c \in [k]$ ▷ each color is of type-1, 2 or 3
7: **for** every $t \in \mathbf{T}$ **do** $R^t \leftarrow \texttt{MNCkC-choose-R}(t)$ ▷ clients in R^t are called *representatives*
8: $S \leftarrow \bigcup_{t \in \mathbf{T}} S^t, R \leftarrow \bigcup_{t \in \mathbf{T}} R^t$
9: guess a pivot p_c for each type-1 or 2 color $c \in [k]$, and i^*_c for each type-3 color $c \in [k]$ such that ▷ p_c's for type-1 and 2 colors c, i^*_c's for all colors $c \in [k]$ are defined in text
- $p_c \in R$ if c is of type-1, $p_c \in S$ if c is of type-2, and
- $i^*_c \in S$ has color c for a type-3 color c.

10: guess a $(1+\epsilon)$-approximate overestimation r_c for $d(i^*_c, p_c)$ for every type-1 or 2 color c
11: **return** $\texttt{MNCkC-clustering-with-pivots}()$ ▷ See Algorithm 3 for its definition

The algorithm is stated in Algorithm 1. In Step 1, we define a set $\mathbf{T}$ of distances depending on the problem, as follows:

- If the norm to minimize is the $\texttt{top-}\ell$ norm, then let $\mathbf{T} = \{t\}$, with t being the ℓ-th largest connection distance in the optimum solution $\mathcal{S}^*$. There are $n|F|$ possibilities for t, so it is affordable to guess its correct value.
- If the norm is a general monotone symmetric norm, then $\mathbf{T} = \left\{(1+\epsilon)^{\lceil \log_{1+\epsilon} d(i,j) \rceil} : i \in F, j \in C\right\}$. Notice that $|\mathbf{T}| = O(\frac{\log n}{\epsilon})$ as we assumed that distances that are not ∞ are integers bounded by $\mathrm{poly}(n)$.

Readers seeking for a more efficient understanding of the core ideas can focus on the $\texttt{top-}\ell$ norm case, where we have only one t in $\mathbf{T}$.

For every $t \in \mathbf{T}$, in Step 3, we obtain a set $\mathcal{S}^t$ of $O\left(\frac{k \ln n}{\epsilon}\right)$ valid stars covering C using Theorem 2.9, and define S^t to be the set of facilities used in $\mathcal{S}^t$ in Step 4. We let i^t_j be the center of the star in $\mathcal{S}^t$ containing j for every $j \in C$, and let $b^t_j = d(j, i^t_j)$ be the connection distance of j in $\mathcal{S}^t$. The properties of the Theorem 2.9 hold with high probability. We assume they are satisfied:

(P) For every $t \in \mathbf{T}$, we have $\sum_{j\in C}((1-\epsilon)b^t_j - t)^+ \leq \sum_{(i^*,J)\in\mathcal{S}^*, j\in J}(d(i^*,j)-t)^+$.

We describe the remaining steps in more detail.

3.1 Steps 5 and 6 of Algorithm 1: Guessing Colors and Types

In Step 5 of Algorithm 1, we randomly choose a function $\texttt{color} : F \to [k]$. With probability $\frac{k!}{k^k}$, the k facilities in $\mathcal{S}^*$ have distinct colors. We assume this happens. For every color $c \in [k]$, let (i^*_c, J^*_c) be the star in $\mathcal{S}^*$ such that i^*_c is of color c. We define $\bar{J}^*_c$ to be the $\lceil \epsilon |J^*_c| \rceil$ clients in J^*_c closest to i^*_c. Let $\tilde{J}^*_{t,c}$ be the $\left\lceil \frac{|\bar{J}^*_c|}{2} \right\rceil$ clients in $\bar{J}^*_c$ with smallest b^t_j values for every $t \in \mathbf{T}$. Notice that i^*_c's, J^*_c's, $\bar{J}^*_c$'s and $\tilde{J}^*_{t,c}$'s are not known to our algorithm after the steps.

We define a *type* for each color $c \in [k]$ as follows:

- If $i^*_c \notin S$, then
  - if $\exists t \in \mathbf{T}, \forall j \in \tilde{J}^*_{t,c}, u_{i^t_j} < k|J^*_c|$, then c is of *type-1a*, otherwise,
  - if $\exists t \in \mathbf{T}, \sum_{j\in\bar{J}^*_c}((1-\epsilon)b^t_j - t)^+ \geq \frac{\epsilon^2}{k} \cdot \sum_{j\in C}((1-\epsilon)b^t_j - t)^+$, then c is of *type-1b*, and otherwise,
  - c is of *type-2*.
- If $i^*_c \in S$, then c is of *type-3*.

Notice that for the type-1a colors, we consider the set $\tilde{J}^*_{t,c}$, but for type-1b colors, we consider the set $\bar{J}^*_c$. We say c is of *type-1* if it is of type-1a or type-1b. Notice the types 1, 2 and 3 partition the set $[k]$ of colors. We guess the types in Step 6 of Algorithm 1. Again, we assume our guesses are correct.

3.2 Step 7 of Algorithm 1: Constructing Representatives R^t

In Step 7 of Algorithm 1, for every $t \in \mathbf{T}$, we construct a *representative set* R^t through the process $\texttt{MNCkC-choose-R}(t)$, described in Algorithm 2.

Algorithm 2 $\texttt{MNCkC-choose-R}(t)$

1: for every $(i, J) \in \mathcal{S}^t$, uniformly sample $\min(|J|, \lceil \frac{2k}{\epsilon}\ln(kn) \rceil)$ clients from J, without replacement; they constitute set R^t_1
2: sample $O\left(\frac{k \ln n}{\epsilon^2}\right)$ clients in $\mathcal{S}^t$, for a sufficiently large hidden constant, each client j being chosen from C with probabilities proportional to $((1-\epsilon)b^t_j - t)^+$, with replacement; they constitute set R^t_2
3: **return** $R^t = R^t_1 \cup R^t_2$

As $|\mathcal{S}^t| = O\left(\frac{k \ln n}{\epsilon}\right)$, we have $|R^t| = O\left(\frac{k^2 \ln^2 n}{\epsilon^2}\right)$ for every $t \in \mathbf{T}$.

In Step 8, we merge all sets $\mathcal{S}^t$ into $\mathcal{S}$ and R^t into R. Therefore, $|\mathcal{S}| = O\left(\frac{k \log^2 n}{\epsilon^2}\right)$ and $|R| = O\left(\frac{k^2 \log^3 n}{\epsilon^3}\right)$.

Lemma 3.1. *With probability at least $1 - \frac{1}{n}$, the following event happens: for every color $c \in [k]$ of type-1, we have $\bar{J}^*_c \cap R \neq \emptyset$.*

Proof. Fix a type-1a color c. Focus on the $t \in \mathbf{T}$ that makes it of type-1a. So, $\forall j \in \tilde{J}^*_{t,c}$, we have $u_{i^t_j} < k|J^*_c|$. As $(i, J) \in \mathcal{S}$ satisfies $|J| \leq u_i$, we have $\Pr[j \in R^t_1] \geq \min\left\{1, \frac{2k\ln(kn)}{\epsilon \cdot u_{i^t_j}}\right\}$ for every $j \in \tilde{J}^*_{t,c}$.

If for some $j \in \tilde{J}^*_{t,c}$ we have $\Pr[j \in R^t_1] = 1$, then $\Pr[\tilde{J}^*_{t,c} \cap R^t_1 \neq \emptyset] = 1$. So, we assume for every $j \in \tilde{J}^*_{t,c}$, we have $\Pr[j \in R^t_1] \geq \frac{2k\ln(kn)}{\epsilon \cdot u_{i^t_j}} \geq \frac{2\ln(kn)}{\epsilon \cdot |J^*_c|} \geq \frac{\ln(kn)}{|\tilde{J}^*_{t,c}|}$. The first inequality holds as $u_{i^t_j} < k|J^*_c|$, and the second inequality follows from that $|\tilde{J}^*_{t,c}| \geq \frac{1}{2} \cdot |\bar{J}^*_c| \geq \frac{\epsilon}{2} \cdot |J^*_c|$.

Notice that the events that $j \in R$ for all $j \in \tilde{J}^*_{t,c}$ are non-positively correlated. So,

$$\Pr[\bar{J}^*_c \cap R^t_1 = \emptyset] \leq \Pr[\tilde{J}^*_{t,c} \cap R^t_1 = \emptyset] \leq \left(1 - \frac{\ln(kn)}{|\tilde{J}^*_{t,c}|}\right)^{|\tilde{J}^*_{t,c}|} \leq e^{-\ln(kn)} = \frac{1}{kn}.$$

Now fix some $c \in [k]$ of type-1b. By the definition of type-1b colors, we have for some $t \in \mathbf{T}$ $\sum_{j \in \bar{J}^*_c} \left((1-\epsilon)b^t_j - t\right)^+ \geq \frac{\epsilon^2}{k} \cdot \sum_{j \in C} \left((1-\epsilon)b^t_j - t\right)^+$. By the way we sample R^t_2 in Step 2 of Algorithm 2, we have $\Pr[\bar{J}^*_c \cap R^t_2 = \emptyset] \leq \frac{1}{n^2}$ if the hidden constant is large enough. The lemma follows from the union bound over the at most k type-1 colors. □

3.3 Step 9 and 10 of Algorithm 1: Guessing Pivots, Optimum Facilities and Radius

In Step 9 of Algorithm 1, we guess the *pivot* p_c for type-1 or 2 colors c, which is defined as follows.

- If c is of type-1, then p_c is defined as any client in $\bar{J}^*_c \cap R$. $p_c \in R \subseteq C$ in this case.
- If c is of type-2, then p_c is defined as the i^t_j for $t \in \mathbf{T}, j \in \tilde{J}^*_{t,c}$ satisfying $u_{i^t_j} \geq k|J^*_c|$ with the smallest b^t_j value. $p_c \in S \subseteq F$ in this case. Such a (t, j) pair exists, since otherwise c would be of type-1a.

We also guess i^*_c for each type-3 color c. By definition, we have $i^*_c \in S$ and thus we can afford this.

In Step 10, we guess $d(i^*_c, p_c)$ for every type-1 or type-2 color c. We can afford to guess a $(1+\epsilon)$-approximation of $d(i^*_c, p_c)$: r_c is the smallest $(1+\epsilon)^z$ that is at most $d(i^*_c, p_c)$ for an integer z.

So, after the guessing, we know the p_c's for type-1 and 2 colors c, and i^*_c's for type-3 colors c. However, we do not know i^*_c's for type-1 and 2 colors c.

3.4 Step 11 of Algorithm 1: Finding Clusters Using Pivots and Radius

With all the information, we can find the optimum clustering. This is done in Step 11 of Algorithm 1, which calls `MNCkC-clustering-with-pivots`() as described in Algorithm 3.

Algorithm 3 `MNCkC-clustering-with-pivots()`

1: $T \leftarrow \emptyset$
2: **for** every color $c \in [k]$ of type-1 **do**
3:     $q_c \leftarrow \arg\max\left\{u_i : i \in \mathtt{ball}_{F \setminus S}(p_c, r_c) \cap \mathtt{color}^{-1}(c)\right\}$; $T \leftarrow T \cup \{q_c\}$
4: **for** every color $c \in [k]$ of type-3 **do**
5:     $T \leftarrow T \cup \{i^*_c\}$
6: **for** every color $c \in [k]$ of type-2 **do**
7:     **if** $p_c \neq i^*_{c'}$ for every type-3 color c' **then** $T \leftarrow T \cup \{p_c\}$
8:     **else** $g_c \leftarrow \arg\max\left\{u_i : i \in \mathtt{ball}_{F \setminus S}(p_c, r_c) \cap \mathtt{color}^{-1}(c)\right\}$; $T \leftarrow T \cup \{g_c\}$
9: **return** T and the best assignment $\sigma \in T^C$ respecting the capacity constraints, either directly (if f is the `top-`ℓ norm) or using Theorem 2.11 (if f is general)

Notice that for a type-2 color c, p_c is not always of color c, so Step 8 in Algorithm 3 is not redundant. Further, we treat T as a set, instead of a multi-set, so $|T| \leq k$ and it may happen that $|T| < k$.

3.5 Analysis of Cost

To analyze the cost of the solution, we explicitly construct an assignment for T with small cost. We build a bipartite graph $H = ([k], T, E_H)$ as follows; it is instructive to correlate the construction with Algorithm 3.

- For every c of type-1, we add (c, q_c) to E_H.
- For every c' of type-3 (which implies $i^*_{c'} \in T$), we add $(c', i^*_{c'})$ to E_H.
- For every c of type-2, we add (c, p_c) to E_H. If additionally $p_c = i^*_{c'}$ for some type-3 color c', then we add (c', g_c) to E_H.

We construct a solution in the following way: Initially, all clients are moved to their corresponding facilities in $\mathcal{S}^*$, with moving cost precisely opt. After that, i^*_c contains $|J^*_c| \le u_{i^*_c}$ clients for any color c. Then, we move the clients from S^* to T according to a "transportation" function $\phi \in \mathbb{Z}^{E_H}_{\ge 0}$: $\phi_{c,i}$ clients will be moved from i^*_c to i. In order for ϕ to be a solution, ϕ must satisfy the following properties:

- $\forall c \in [k], \sum_{(c,i)\in E_H} \phi_{c,i} = |J^*_c|$;
- $\forall i \in T, \sum_{(c,i)\in E_H} \phi_{c,i} \le u_i$.

In the following, we construct ϕ. Connected components in H are only of the following three possibilities, and we construct ϕ for each of them (See Figure 1 for an illustration):

(1) Some type-1 color c connected to $q_c \in T$. Then q_c has degree 1 and $\{c, q_c\}$ is maximally connected. We set $\phi_{c,q_c} = |J^*_c|$. We have $u_{q_c} \ge u_{i^*_c} \ge |J^*_c|$.

(2) Several type-2 colors and no type-3 color connect to facility $i \in T$. Let $D := \{c : p_c = i\}$, then $D \cup \{i\}$ is maximally connected. Set $\phi_{c,i} = |J^*_c|$ for every $c \in D$. We have $\sum_{c\in D} |J^*_c| \le \sum_{c\in D} \frac{u_i}{k} \le u_i$ as all colors in D are of type-2.

(3) Facility $i = i^*_{c'} \in T \cap S^*$ is connected to one type-3 color c'. Some type-2 colors may also connect to i. Let $D := \{c : p_c = i, c \text{ is of type-2}\}$; D may be empty. $\{c', i\} \cup D \cup \{g_c \mid c \in D\}$ is one connected component in H. Similar to (ii), $\sum_{c\in D} |J^*_c| \le u_i$. We define $\phi_{c,i} = |J^*_c|$ for every $c \in D$. We then define $\phi_{c',i} = \min\{|J^*_{c'}|, u_i - \sum_{c\in D} |J^*_c|\}$. If $u_i - \sum_{c\in D} |J^*_c| < |J^*_{c'}|$, then we define ϕ_{c',g_c} values for all $c \in D$ so that $\phi_{c',g_c} \le |J^*_c|$ for every $c \in D$, and $\sum_{c\in D} \phi_{c',g_c} = |J^*_{c'}| - \phi_{c',i}$. This is feasible since $\sum_{c\in D} |J^*_c| = u_i - \phi_{c',i} \ge |J^*_{c'}| - \phi_{c',i}$. As for $c \in D$, i^*_c is a candidate for g_c, $u_{g_c} \ge u_{i^*_c} \ge |J^*_c|$, so ϕ does not violate the capacity constraint for g_c.

In summary, the two properties needed for the transportation function ϕ are satisfied, which implies the assignment of C to T respects all capacity constraints.

We then bound the f-norm cost of moving clients from S^* to T using the transportation function ϕ. For each client j, let β_j be the moving distance of this step, and b^*_j be the connection distance of j in the optimum solution $\mathcal{S}^*$. The main lemma we prove is the following:

Lemma 3.2. *For every $t \in \mathbf{T}$, we have*

$$\sum_{j\in C} (\beta_j - (2 + O(\epsilon))t)^+ \le (2 + O(\epsilon)) \sum_{j\in C} (b^*_j - t)^+.$$

Proof. Throughout the proof, we fix a $t \in \mathbf{T}$. We distinguish different color types to prove the inequality.

For a type-1 color c, J^*_c will be moved from i^*_c to q_c.

$$d(i^*_c, q_c) \le d(i^*_c, p_c) + d(p_c, q_c) \le (2 + \epsilon) d(i^*_c, p_c) \le \frac{2+\epsilon}{1-\epsilon} \cdot \frac{1}{|J^*_c|} \sum_{j\in J^*_c} d(i^*_c, j) = \frac{2+\epsilon}{1-\epsilon} \cdot \frac{1}{|J^*_c|} \sum_{j\in J^*_c} b^*_j.$$

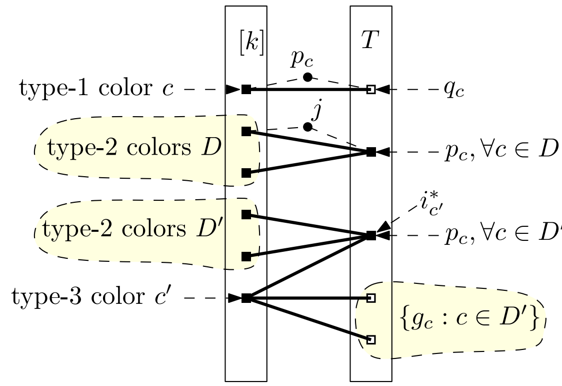

Figure 1: 3 types of connected components of H. In the first type, a type-1 color c is connected to q_c. Notice that q_c is defined via a representative client $p_c \in R \cap \bar{J}^*_c$. In the second type, we have many type-2 colors $c \in D$ connected to their common p_c. p_c for a type-2 color c is defined via a client j. In the third type, we have many type-2 colors $c \in D$ (D' in the figure to avoid confusion) connected to their common p_c, which is $i^*_{c'}$ for a type-3 color c'. Then we have edges $(c', i^*_{c'})$ and (c', g_c) for every $c \in D$ in the component. Solid squares in T are in S, and empty squares in T are not in S.

The second inequality is by that $r_c \leq (1+\epsilon)d(i^*_c, p_c)$ and $q_c \in \texttt{ball}_{F\setminus S}(p_c, r_c)$. The third inequality is by that $p_c \in \bar{J}^*_c$ and $\bar{J}^*_c$ are the $\lceil \epsilon |J^*_c| \rceil$ clients j in J^*_c with the smallest $d(i^*_c, j)$ values. So,

$$\sum_{j \in J^*_c} \left(\beta_j - \frac{2+\epsilon}{1-\epsilon} \cdot t\right)^+ = |J^*_c| \cdot \left(d(i^*_c, q_c) - \frac{2+\epsilon}{1-\epsilon} \cdot t\right)^+ = \left(|J^*_c| \cdot d(i^*_c, q_c) - \frac{2+\epsilon}{1-\epsilon} \cdot |J^*_c| \cdot t\right)^+$$
$$\leq \left(\frac{2+\epsilon}{1-\epsilon} \cdot \sum_{j \in J^*_c} b^*_j - \frac{2+\epsilon}{1-\epsilon} \cdot |J^*_c| \cdot t\right)^+ \leq \frac{2+\epsilon}{1+\epsilon} \cdot \sum_{j \in J^*_c} (b^*_j - t)^+ = (2 + O(\epsilon)) \sum_{j \in J^*_c} (b^*_j - t)^+. \tag{1}$$

We consider type-2 and type-3 colors together. For the case (ii) above, we moved $|J^*_c|$ clients from i^*_c to i for every $c \in D$. For the case (iii), we did the same; additionally, for every $c \in D$, we moved at most $|J^*_c|$ clients from $i^*_{c'} = p_c$ to g_c. Notice that by the definition of g_c for every $c \in D$, we have $d(i^*_{c'}, g_c) = d(p_c, g_c) \leq r_c \leq (1+\epsilon)d(p_c, i^*_c)$. Some clients in $i^*_{c'}$ stayed at $i = i^*_{c'}$. Therefore, we have

$$\sum_{c \text{ type-2 or } 3, j \in J^*_c} \left(\beta_j - \frac{(1+\epsilon)(2-\epsilon)}{1-\epsilon} \cdot t\right)^+ \leq 2 \sum_{c \text{ type-2}} |J^*_c| \cdot \left((1+\epsilon) \cdot d(i^*_c, p_c) - \frac{(1+\epsilon)(2-\epsilon)}{1-\epsilon} \cdot t\right)^+$$
$$= 2(1+\epsilon) \sum_{c \text{ type-2}} |J^*_c| \cdot \left(d(i^*_c, p_c) - \frac{2-\epsilon}{1-\epsilon} \cdot t\right)^+. \tag{2}$$

Now, we focus on each type-2 color c. By the definition of p_c, we have $p_c = i^t_{j'}$ for some $j' \in \tilde{J}^*_{t,c}$. As c is not of type 1, $\sum_{j \in \bar{J}^*_c} ((1-\epsilon)b^t_j - t)^+ < \frac{\epsilon^2}{k} \cdot \sum_{j \in C} ((1-\epsilon)b^t_j - t)^+$. We have

$$\left(d(j', p_c) - \frac{1}{1-\epsilon} \cdot t\right)^+ = \left(b^t_{j'} - \frac{1}{1-\epsilon} \cdot t\right)^+ \leq \frac{2}{|\bar{J}^*|} \sum_{j \in \bar{J}^*_c} \left(b^t_j - \frac{1}{1-\epsilon} \cdot t\right)^+$$
$$= \frac{2}{(1-\epsilon)|\bar{J}^*_c|} \sum_{j \in \bar{J}^*_c} \left((1-\epsilon)b^t_j - t\right)^+ \leq \frac{2\epsilon^2}{(1-\epsilon)k|\bar{J}^*_c|} \cdot \sum_{j \in C} \left((1-\epsilon)b^t_j - t\right)^+$$
$$\leq \frac{2\epsilon}{(1-\epsilon)k|J^*_c|} \sum_{j \in C} \left((1-\epsilon)b^t_j - t\right)^+ = \frac{O(\epsilon)}{k|J^*_c|} \sum_{j \in C} \left((1-\epsilon)b^t_j - t\right)^+.$$

The first inequality used that $j' \in \tilde{J}^*_{t,c}$ and $\tilde{J}^*_{t,c}$ contains the $\left\lceil \frac{|\bar{J}^*_c|}{2} \right\rceil$ clients in $\bar{J}^*_c$ with the smallest b^t_j values. The third inequality used that $|\bar{J}^*_c| \geq \epsilon |J^*_c|$.

As $j' \in \tilde{J}^*_{t,c} \subseteq \bar{J}^*_c$, we have

$$(d(i^*_c, j') - t)^+ \leq \frac{1}{1-\epsilon} \cdot \frac{1}{|J^*_c|} \sum_{j \in J^*_c} (d(i^*_c, j) - t)^+ = \frac{1 + O(\epsilon)}{|J^*_c|} \sum_{j \in J^*_c} (b^*_j - t)^+.$$

Combining the above two inequalities, we have

$$\begin{aligned} |J^*_c| \cdot \Big(d(i^*_c, p_c) - \frac{2-\epsilon}{1-\epsilon} \cdot t\Big)^+ &\leq |J^*_c| \cdot \Big(d(i^*_c, j') - t\Big)^+ + |J^*_c| \cdot \Big(d(j', p_c) - \frac{1}{1-\epsilon} \cdot t\Big)^+ \\ &\leq (1 + O(\epsilon)) \sum_{j \in J^*_c} (b^*_j - t)^+ + \frac{O(\epsilon)}{k} \sum_{j \in C} ((1-\epsilon) b^t_j - t)^+. \end{aligned}$$

Applying the upper bound to (2), we get

$$\begin{aligned} &\sum_{c \text{ type-2 or } 3, j \in J^*_c} \Big(\beta_j - \frac{(1+\epsilon)(2-\epsilon)}{1-\epsilon} \cdot t\Big)^+ \\ &\leq (2 + O(\epsilon)) \cdot \sum_{c \text{ type } 2} \sum_{j \in J^*_c} (b^*_j - t)^+ + O(\epsilon) \cdot \sum_{j \in C} \big((1-\epsilon) b^t_j - t\big)^+. \end{aligned} \tag{3}$$

Summing up contribution for colors all the 3 types, by adding (1) and (3), we get

$$\begin{aligned} &\sum_{j \in C} (\beta_j - (2 + O(\epsilon))t)^+ = \sum_{c \in [k]} \sum_{j \in J^*_c} (\beta_j - (2 + O(\epsilon))t)^+ \\ &\leq (2 + O(\epsilon)) \sum_{j \in C} (b^*_j - t)^+ + O(\epsilon) \sum_{j \in C} ((1-\epsilon) b^t_j - t)^+ \leq (2 + O(\epsilon)) \sum_{j \in C} (b^*_j - t)^+. \end{aligned}$$

The last inequality used $\sum_{j \in C} ((1-\epsilon) b^t_j - t)^+ \leq \sum_{j \in C} (b^*_j - t)^+$, which is from Property (P). □

Recall that in our constructed solution, we moved clients from C to T in two steps. We first moved C to S^*, which incur a cost of opt. Then we moved clients from S^* to T, with moving distance vector β.

Now we describe the $f \equiv \texttt{top-}\ell$ and general f case separately. For the case $f \equiv \texttt{top-}\ell$, Lemma 3.2 holds for the ℓ-th largest coordinate of b^*. By Lemma 2.4, the moving cost of the second step is

$$\begin{aligned} |\beta|_{\texttt{top-}\ell} &\leq \ell \cdot (2 + O(\epsilon))t + \sum_{j \in C} (\beta_j - (2 + O(\epsilon))t)^+ \leq (2 + O(\epsilon))\ell t + (2 + O(\epsilon)) \sum_{j \in C} (b^*_j - t)^+ \\ &\leq (2 + O(\epsilon)) \left(\ell t + \sum_{j \in C} (b^*_j - t)^+\right) = (2 + O(\epsilon)) \cdot |b^*|_{\texttt{top-}\ell} = (2 + O(\epsilon))\text{opt}. \end{aligned}$$

This implies that $\texttt{top-}\ell$ cost of our constructed solution is at most $(3 + O(\epsilon)) \cdot \text{opt}$.

Now consider a general monotone symmetric norm f. We show that for every integer $\ell \in [n]$, we have $|\beta|_{\texttt{top-}\ell} \leq (2 + O(\epsilon))|b^*|_{\texttt{top-}\ell}$. By Lemma 2.6, this implies $|\beta|_f \leq (2 + O(\epsilon))|b^*|_f = (2 + O(\epsilon))\text{opt}$. Focus on any $\ell \in [n]$ and let t be the ℓ-th largest coordinate in b^*. Then, we have $t' = (1+\epsilon)^{\lceil \log_{1+\epsilon} t \rceil} \in \mathbf{T}$. Applying Lemma 3.2 with t being this t' will prove $|\beta|_{\texttt{top-}\ell} \leq (2 + O(\epsilon))|b^*|_{\texttt{top-}\ell}$. Therefore, the f-norm cost our constructed solution is at most $(3 + O(\epsilon)) \cdot \text{opt}$. In both cases, scaling ϵ at the beginning gives us a $(3 + \epsilon)$-approximation.

3.6 Analysis of Runtime

Now we analyze the running time of our algorithm. When all guesses are correct, the coloring satisfies our requirement, and the events mentioned in Lemma 3.1 happen, the algorithm returns a solution with value at most $(3 + O(\epsilon))\mathrm{opt}$.

For the runtime, all steps except coloring and guessing are polynomial. With probability $\frac{k!}{k^k}$, the color function in Step 5 satisfies our requirement, so we need to run the algorithm $O(k^k \ln n)$ times to boost the success probability to $1 - \frac{1}{n}$. For the guessing part, Step 6 has 3^k possible choices for the types, Step 9 has $(|R| + |S|)^k = (\frac{k \ln n}{\epsilon})^{O(k)}$ choices for pivots, and finally, as we can always assume $d(c, f) = \mathrm{poly}(n)$ for all $c \in C$ and $f \in F$, Step 10 has $(\log_{1+\epsilon} \mathrm{poly}(n))^k = \left(\frac{\ln n}{\epsilon}\right)^{O(k)}$ choices for the radius. In total, the running time is $(\frac{k \ln n}{\epsilon})^{O(k)} n^{O(1)}$, which can be bounded by $(\frac{k}{\epsilon})^{O(k)} n^{O(1)}$. [4]

4 FPT Time $(1+\frac{2}{ec}+\epsilon)$-Approximation for Top-$cn$ Norm k-Clustering

In this section, we give the tight $(1 + \frac{2}{ec} + \epsilon)$-approximation algorithm for the $\texttt{top-}cn$ norm k-clustering problem for $c \in (\frac{1}{e}, 1]$, proving Theorem 1.3. Notice that the case $c \le \frac{1}{e}$ has an approximation ratio of $3 + \epsilon$, as stated in Corollary 2.10.

4.1 Useful Tools for Top-ℓ Norms

In this section, we describe some useful tools. We assume the distances in the metric d are non-negative integers. It will be convenient to use the *occurrence-times vectors* to represent distance vectors: for each $a \in \mathbb{Z}_{\ge 0}$, we have a coordinate indicating the number of times a appear in the distance vector; that is, how many clients have distance a to their nearest facility. With this in mind, we define the set of occurrence-times vectors to be

$$\mathcal{S}_n := \{\delta \in \mathbb{R}_{\ge 0}^{\mathbb{Z}_{\ge 0}} : |\delta|_1 = n \text{ and } \delta \text{ has a finite support}\}.$$

For a $\delta \in \mathcal{S}_n$ and $a \in \mathbb{Z}_{\ge 0}$, we can think of δ_a as the number of times a appear in the distance vector. It is convenient to allow fractional occurrence times.

For any vector $\phi \in \mathbb{R}_{\ge 0}^{Z \ge 0}$ with finite support, we define $\overline{L_1}(\phi) = \sum_{a=0}^{\infty} \phi_a \cdot a$. (Throughout, we shall typically use $\overline{f}$ to denote a norm function f using the occurrence-times vector representation.)

Definition 4.1. *For every $\delta \in \mathcal{S}_n$ and real $\ell \in [0, n]$, we define $\overline{\texttt{top-}\ell}(\delta)$ to be the value of the following linear program with variables $\alpha \in \mathbb{R}_{\ge 0}^{\mathbb{Z}_{\ge 0}}$: maximize $\overline{L_1}(\alpha)$ subject to $0 \le \alpha \le \delta$ and $|\alpha|_1 = \ell$.*

To get some intuition about the definition, consider the case where d is a distance vector of dimension n and δ is its correspondent occurrence-times vector. Then $\overline{\texttt{top-}\ell}(\delta) = |d|_{\texttt{top-}\ell}$. We extended the definition to real vectors δ.

Lemma 4.2. *For every real $\ell \in [0, n]$, we have that $\overline{\texttt{top-}\ell}(\cdot)$ concave over $\mathcal{S}_n$.*

This is in contrast to the $\texttt{top-}\ell$ norm function using the normal representation of vectors, which is convex.

Definition 4.3. *We say a vector $\delta \in \mathcal{S}_n$ dominates a vector $\delta' \in \mathcal{S}_n$ with a factor of γ for some real $\gamma \ge 0$, denoted as $\delta' \preceq_\gamma \delta$, if $\overline{\texttt{top-}\ell}(\delta') \le \gamma \cdot \overline{\texttt{top-}\ell}(\delta)$ for every real $\ell \in [0, n]$. We use $\preceq$ for $\preceq_1$.*

Lemma 4.4. *Assume $\delta^1, \delta^2, \cdots, \delta^H, \delta'^1, \delta'^2, \cdots, \delta'^H$ are $2H$ vectors in $\mathcal{S}_n$, $\gamma \in \mathbb{R}_{>0}$, and $\delta'^h \preceq_\gamma \delta^h$ for every $h \in [H]$. Let $\beta_1, \beta_2, \cdots, \beta_H \in [0, 1]$ satisfy $\sum_{h=1}^H \beta_h = 1$. Then*

$$\sum_{h=1}^{H} \beta_h {\delta'}^h \preceq_\gamma \sum_{h=1}^{H} \beta_h \delta^h.$$

[4] We need to bound $(\ln n)^{O(k)}$. If $k \le \sqrt{\log n}$, then this is upper bounded by $n^{O(1)}$. Otherwise, it is upper bounded by $k^{O(k)}$.

Corollary 4.5. *Let $\delta \in \mathcal{S}_n$, and $\delta = \sum_{h=0}^{H} \theta^h$ where $H \in \mathbb{Z}_{\geq 0}$ and $\theta^0, \theta^1, \cdots, \theta^H \geq 0$. Let $\gamma \geq 1$. For every $h \in [H]$, let $b_h \in \mathbb{Z}_{\geq 0}$ satisfy $b_h \leq \gamma \cdot \frac{\overline{L_1}(\theta^h)}{|\theta^h|_1}$. Let $\delta' = \theta^0 + \sum_{h=1}^{H} |\theta^h|_1 \cdot \mathbf{e}_{b_h}$, where $\mathbf{e}_b$ for any $b \in \mathbb{Z}_{\geq 0}$ is the vector in $\mathbb{R}_{\geq 0}^{Z_{\geq 0}}$ with $(\mathbf{e}_b)_b = 1$ and $(\mathbf{e}_b)_{b'} = 0$ if $b' \neq b$.*

Then $\delta' \preceq_\gamma \delta$.

We treat δ as n fractional values. Each θ^h contains a disjoint portion of δ. For every $h \in H$, we replace θ^h with $|\theta^h|_1$ fractional values equaling to b_h. The lemma says that the resulting occurrence-times vector is dominated by the original one with a factor of γ.

Definition 4.6. *For any two vectors $\delta, \delta' \in \mathcal{S}_n$, we say some $\phi \in \mathcal{S}_n$ can be obtained by adding δ and δ' if there exists some $z \in \mathbb{R}_{\geq 0}^{\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}}$ such that*

- *$\sum_{a'} z_{a,a'} = \delta_a$ for every $a \in \mathbb{Z}_{\geq 0}$,*
- *$\sum_{a} z_{a,a'} = \delta'_{a'}$ for every $a' \in \mathbb{Z}_{\geq 0}$, and*
- *$\phi_t = \sum_{a+a'=t} z_{a,a'}$ for every $t \in \mathbb{Z}_{\geq 0}$.*

z gives a matching between the vectors δ and δ'. Then the occurrence-time vector ϕ is obtained by adding the distance vectors for δ and δ', using the matching z.

Definition 4.7. *Let $\delta, \delta' \in \mathcal{S}_n$, and $z \in \mathbb{R}_{\geq 0}^{\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}}$ be the unique vector satisfying the first two conditions in Definition 4.6 and the following condition:*

- *There are no integers $a < a', b > b'$ such that $z_{a,b} > 0$ and $z_{a',b'} > 0$.*

Then, we use $\delta \oplus \delta'$ to denote the vector $\phi \in \mathcal{S}_n$ satisfying the third condition of Definition 4.6. We use $\delta \otimes 2$ to denote $\delta \oplus \delta$.

Lemma 4.8. *Let $\delta, \delta' \in \mathcal{S}_n$, $\phi = \delta \oplus \delta'$, $\phi' \in \mathcal{S}_n$ be obtained by adding δ and δ'. Then $\phi' \preceq \phi$.*

Theorem 4.9. *Let $\delta, \delta' \in \mathcal{S}_n$ such that $\delta' \preceq_\gamma \delta$ for some $\gamma > 0$. Let $c \in (\frac{1}{e}, 1]$ and $\alpha \in [0, c]$. Then we have*

$$\overline{\texttt{top-}cn}\big((1-\alpha)\delta + \alpha \cdot (\delta \oplus \delta' \otimes 2)\big) \leq \left(1 + \frac{2\alpha\gamma}{c}\right) \overline{\texttt{top-}cn}(\delta).$$

We only apply the theorem for $\alpha = \frac{1}{e}$ and $\gamma = 1 + O(\epsilon)$.

4.2 The Algorithm

Throughout this and the next section, given any set $S \subseteq F$ (it is possible that $|S| \neq k$), we shall use $\vec{d}(S)$ to denote the vector $(d(j, S))_{j \in C}$ of n distances; we call it the distance vector for S. So the goal of the $\texttt{top-}cn$ norm k-clustering problem is to minimize $|\vec{d}(S)|_{\texttt{top-}cn}$ subject to $S \subseteq F, |S| = k$.

Let S^* be the unknown set of k facilities in the optimum solution. We let $\text{opt} = |\vec{d}(S^*)|_{\texttt{top-}cn}$. For every $i^* \in S^*$, and the set $J \subseteq C$ of clients connected to i^* in the optimum solution, we refer to (i^*, J) as an *optimum cluster*.

Definition 4.10 (Per-Client Costs and Cores)**.** *For every optimum cluster (i^*, J), we define its* per-client cost *to be $\frac{1}{|J|} \sum_{j \in J} d(i^*, j)$. We define its* core *to be the set of the $\lceil \epsilon |J| \rceil$ clients in J with the smallest $d(i^*, j)$ values.*

The pseudo-code for the algorithm is given in Algorithm 4. In Step 2, we apply Theorem 2.9 to obtain a solution $\mathcal{S}$ and let S be the set of open facilities. As there are no capacities, we simply use S to denote the solution and discard the notion $\mathcal{S}$. We have $|S| \leq \frac{O(k \log n)}{\epsilon}$. Moreover, $|\vec{d}(S)|_{\texttt{top-}cn} \leq \frac{\text{opt}}{1-\epsilon}$.

We describe the remaining steps of the algorithm in more detail.

Algorithm 4 Top-cn Norm k-Clustering

1: obtain a set $\mathcal{S}$ of valid stars using Theorem 2.9 with $h \equiv \texttt{top-}cn$ and $u_i = \infty, \forall i \in F$
2: $S \leftarrow \{i \in F : \exists J, (i, J) \in \mathcal{S}\}$
3: $R \leftarrow$ `TpcnC-choose-R`() ▷ Clients in R are called *representatives*
4: guess a multi-set $P \subseteq S \cup R$ of size k ▷ Points in P are called *pivots*
5: guess a vector $(r_p)_{p \in P}$ where each r_p is 0 or an integer power of $(1+\epsilon)$ in $[1, \Delta]$
6: **return** `TpcnC-clustering-with-pivots`()

Algorithm 5 `TpcnC-choose-R`()

$R \leftarrow \emptyset$
repeat k times:
  randomly choose a client $j \in C$, with probabilities proportional to $d(j, S)$
  $R \leftarrow R \cup \{j\}$
return R

4.3 Step 3 of Algorithm 4: Choosing Representatives

In this step, we call the procedure `TpcnC-choose-R`(), described in Algorithm 5. Before analyzing the properties of the representative set R, we make some definitions and partition the optimum clusters into 3 types: Given an optimum cluster (i^*, J) with core J', we say (i^*, J) is of

- *type-1* if $\sum_{j \in J'} d(j, S) \geq \frac{\epsilon^3}{k} \cdot |\vec{d}(S)|_1$,
- *type-2* if $\sum_{j \in J'} d(j, S) < \frac{\epsilon^3}{k} \cdot |\vec{d}(S)|_1$ and $\sum_{j \in J'} d(i^*, j) \geq \frac{\epsilon^2}{k} \cdot |\vec{d}(S)|_1$, and
- *type-3* if $\sum_{j \in J'} d(j, S) < \frac{\epsilon^3}{k} \cdot |\vec{d}(S)|_1$ and $\sum_{j \in J'} d(i^*, j) < \frac{\epsilon^2}{k} \cdot |\vec{d}(S)|_1$.

Notice that the three types partition all the optimum clusters.

Lemma 4.11. *With probability at least $\frac{\epsilon^{3k}}{k^k}$, the following event happens: For every optimum cluster (i^*, J) of type-1 with core J', we have $J' \cap R \neq \emptyset$. In other words, R intersects the core of every type-1 optimum cluster.*

Proof. Focus on the process of choosing one j with probability proportional to $d(j, S)$. For the core J' of a type-1 optimum cluster (i^*, J), we have

$$\Pr[j \in J'] = \frac{\sum_{j' \in J'} d(j', S)}{\sum_{j' \in C} d(j', S)} \geq \frac{\epsilon^3}{k},$$

Let $k' \leq k$ be the number of type-1 optimum clusters. We focus on the first k' iterations of the repeat loop of Algorithm 5, and use $j_1, j_2, \cdots, j_{k'}$ to denote the clients chosen in the k' iterations. Fix any permutation $(i^*_1, J_1), (i^*_2, J_2), \cdots, (i^*_{k'}, J_{k'})$ of the k' clusters. The probability that R hits the cores of all type-1 optimum clusters is at least

$$\Pr[\forall o \in [k'] : j_o \text{ is in core of } (i^*_o, A_o)] \geq \left(\frac{\epsilon^3}{k}\right)^{k'} \geq \left(\frac{\epsilon^3}{k}\right)^{k} = \frac{\epsilon^{3k}}{k^k}.$$ □

From now on, we assume Step 3 of Algorithm 4 is successful, which means the event in Lemma 4.11 occurs.

Lemma 4.12. *For every type-1 or 2 optimum cluster (i^*, J) with per-client cost $\bar{d}$, we have $d(i^*, R \cup S) \leq \frac{\bar{d}}{1-\epsilon}$.*

Proof. Let J' be the core of (i^*, J). First, assume (i^*, J) is of type-1. As Step 3 is successful, we have $J' \cap R \neq \emptyset$. Let p be any client in $J' \cap R$. For any $j \in (J \setminus J') \cup \{p\}$, we have $d(i^*, j) \geq d(i^*, p)$. So

$$|J|\bar{d} = \sum_{j \in J} d(j, i^*) \geq \sum_{j \in (J \setminus J') \cup \{p\}} d(j, i^*) \geq (|J| - \lceil \epsilon |J| \rceil + 1) d(i^*, p) \geq (|J| - \epsilon |J|) d(i^*, p).$$

Therefore, we have $d(i^*, p) \leq \frac{|J|\bar{d}}{|J| - \epsilon|J|} = \frac{\bar{d}}{1-\epsilon}$.

Then, assume (i^*, J) is of type-2. By the definition of type-2, we have $\sum_{j \in J'} d(j, S) < \epsilon \sum_{j \in J'} d(i^*, j)$. Consider the facility $p \in S$ that is closest to i^*.

$$\sum_{j \in J'} d(i^*, p) \leq \sum_{j \in J'} \big(d(i^*, j) + d(j, S)\big) = \sum_{j \in J'} d(i^*, j) + \sum_{j \in J'} d(j, S) \leq (1+\epsilon) \sum_{j \in J'} d(i^*, j).$$

So,

$$d(i^*, p) \leq (1+\epsilon) \cdot \frac{\sum_{j \in J'} d(i^*, j)}{|J'|} \leq (1+\epsilon) \cdot \frac{\sum_{j \in J} d(i^*, j)}{|J|} = (1+\epsilon)\bar{d} \leq \frac{\bar{d}}{1-\epsilon}.$$

The second inequality used that $d(i^*, j) \leq d(i^*, j')$ for every $j \in J'$ and $j' \in J \setminus J'$. □

4.4 Steps 4 and 5 of Algorithm 4: Guessing Pivots and Radius Vector

In Steps 4 and 5 of Algorithm 4, we guess a multi-set P of k pivots from $R \cup S$, and a radius r_p for every $p \in P$. There is a *desired* pivot p for every optimum cluster (i^*, J), and the pivot p has a *desired* radius. Step 4 is successful if P is the set of k desired pivots for the k optimum clusters, and Step 5 is successful if r_p for each $p \in P$ is the desired radius for p. For every optimum cluster (i^*, J), the desired pivot p for the cluster and its desired radius are defined as follows:

- If (i^*, J) is of type-1 or 2, then $d(i^*, S \cup R) \leq \frac{\bar{d}}{1-\epsilon}$ by Lemma 4.12. The desired pivot p is the closest point in $S \cup R$ to i^*. Thus, we have $d(i^*, p) \leq \frac{\bar{d}}{1-\epsilon}$. The desired radius is $d(i^*, p)$ rounded up to the nearest integer power of $1+\epsilon$. Therefore, if r_p is the desired radius, we have $r_p \leq \frac{1+\epsilon}{1-\epsilon}\bar{d}$.
- If (i^*, J) is of type-3, then the desired pivot p is the closest facility in S to i^*, and its desired radius is 0.

Again, we assume the two steps are successful from now on. Therefore, P is the set of desired pivots for the optimum clusters and r_p for each $p \in P$ is the desired radius for p.

Lemma 4.13. *There is a set S'^* of k facilities, one from $\mathtt{ball}_F(p, r_p)$ for each $p \in P$, and an assignment vector $\sigma \in (S'^*)^C$ such that the following properties hold.*

(4.13a) $\left| \big(d(j, \sigma_j)\big)_{j \in C} \right|_{\mathtt{top}\text{-}cn} \leq (1 + O(\epsilon)) \cdot \mathrm{opt}.$

(4.13b) For every $p \in P$, and the facility $i'^* \in S'^*$ in $\mathtt{ball}_F(p, r_p)$, we have

$$r_p \leq \frac{1+\epsilon}{1-\epsilon} \cdot \frac{\sum_{j \in \sigma^{-1}(i'^*)} d(i'^*, j)}{|\sigma^{-1}(i'^*)|}.$$

We remark that the balls $\mathtt{ball}_F(p, r_p)$ may overlap with each other and thus one facility in P may be in two different balls. However, we guarantee that there is precisely one facility in S'^* that is designated to $\mathtt{ball}_F(p, r_p)$ for each $p \in P$, and the facility is inside the ball.

Proof of Lemma 4.13. We construct the solution S'^* with the assignment σ as follows. For every optimum cluster (i^*, J) with per-client-cost $\bar{d}$, we include a facility i'^* in S'^*. Let p be the desired pivot for this optimum cluster. If (i^*, J) is of type-1 or 2, we let $i'^* = i^*$. In this type, we have $i'^* = i^* \in \texttt{ball}_F(p, r_p)$ as r_p is the desired radius for p. Otherwise, we are of type-3 and the pivot p is the nearest facility in S to i^*. We let $i'^* = p$ and thus $i'^* \in \texttt{ball}(p, r_p = 0)$. In any type, we let $\sigma_j = i'^*$ for every $j \in J$. b holds if (i^*, J) is of type-1 or 2 as $r_p \leq \frac{1+\epsilon}{1-\epsilon}\bar{d}$. of type-3, we have $r_p = 0$ and the property holds trivially.

It remains to show a. Notice that the difference between the left side of the inequality in a and opt come from type-3 optimum clusters. Therefore, for every type-1 or 2 optimum cluster (i^*, J), we define $\rho_j = 0$ for every $j \in J$. Fix a type-3 optimum cluster (i^*, J), and its correspondent desired pivot p, we define $\rho_j = (d(j,p) - d(j, i^*))^+$ for every $j \in J$.

$$\sum_{j \in J} \rho_j \leq |J| \cdot d(i^*, p) \leq |J| \cdot \frac{1}{|J'|}\left(\frac{\epsilon^3}{k} + \frac{\epsilon^2}{k}\right)|\vec{d}(S)|_1 \leq \frac{1}{\epsilon} \cdot \frac{2\epsilon^2}{k}|\vec{d}(S)|_1 = \frac{2\epsilon}{k}|\vec{d}(S)|_1.$$

The second inequality comes from triangle inequality and the definition of type-3.

Summing up over all type-3 optimum clusters, we have $|\rho|_1 \leq 2\epsilon|\vec{d}(S)|_1$. Therefore,

$$|\rho|_{\texttt{top-}cn} \leq |\rho|_1 \leq O(\epsilon) \cdot |\vec{d}(S)|_1 \leq O(\epsilon) \cdot |\vec{d}(S)|_{\texttt{top-}cn} \leq O(\epsilon) \cdot \text{opt}.$$

The third inequality used that $c \in (\frac{1}{e}, 1]$. □

4.5 Step 6 of Algorithm 4: Find Clustering using P and $(r_p)_{p \in P}$

In this final step, we try to find a clustering using the pivot set P and the radius vector $(r_p)_{p\in P}$ as a guide, by calling `TpcnC-clustering-with-pivots`$(P, (r_p)_{p\in P})$ in Algorithm 6. The existence of a good clustering is guaranteed by Lemma 4.13.

Algorithm 6 `TpcnC-clustering-with-pivots()`

1: solve program (4) to obtain solution $\left((x_{ij}^{(p)})_{p,i,j}, (y_i^{(p)})_{p,i}, \delta \in \mathcal{S}_n\right)$
2: **for** every $p \in P$ **do** randomly open a facility i, with probabilities $y_i^{(p)}$
3: **return** the set of open facilities

In Step 1 of Algorithm 6, we solve the following program:

$$\min \qquad \overline{\texttt{top-}cn}(\delta) \tag{4}$$

$$\sum_{i \in \texttt{ball}_F(p, r_p)} y_i^{(p)} = 1 \qquad \forall p \in P \tag{5}$$

$$x_{ij}^{(p)} \leq y_i^{(p)} \qquad \forall p \in P, i \in \texttt{ball}_F(p, r_p), j \in C \tag{6}$$

$$\sum_{p \in P, i \in \texttt{ball}(p, r_p)} x_{ij}^{(p)} = 1 \qquad \forall j \in C \tag{7}$$

$$\frac{1+\epsilon}{1-\epsilon}\sum_{j\in C} d(i,j)x_{ij}^{(p)} - r_p \sum_{j \in C} x_{ij}^{(p)} \geq 0 \qquad \forall p \in P, i \in \texttt{ball}_F(p, r_p) \tag{8}$$

$$\delta_a - \sum_{p,i,j:d(i,j)=a} x_{ij}^{(p)} = 0 \qquad \forall a \in [0, \Delta] \tag{9}$$

$$x_{ij}^{(p)}, y_i^{(p)} \geq 0 \qquad \forall p \in P, i \in F, j \in C \tag{10}$$

We now focus on the correspondent integer program to program (4), whose goal is to find the set S'^* and the assignment σ in Lemma 4.13. $y_i^{(p)}$ indicates whether i is the open facility in $\mathtt{ball}_F(p, r_p)$, and $x_{ij}^{(p)}$ indicates whether the client j is connected to facility i in $\mathtt{ball}_F(p, r_p)$. (5) indicates we open exactly one facility in $\mathtt{ball}_F(p, r_p)$, (6) indicates that a client can only be connected to an open facility, and (7) requires every client to be connected to a facility in all balls. (8) is due to b, and (9) gives the definition of $\delta \in \mathcal{S}_n$.

We then describe the objective (4), which is not a linear function of δ. By Lemma 2.4, we know that $\overline{\mathtt{top}\text{-}cn}(\delta) = \min_{t\in[0,\Delta]} \left(\sum_a (a-t)^+\delta_a + cnt\right)$, we can solve the program by enumerating integers $t \in [0, \Delta]$. [5]

After solving the CP(4), we obtain the solution $\left((x_{ij}^{(p)})_{p,i,j}, (y_i^{(p)})_{p,i}, \delta \in \mathcal{S}_n\right)$. Let $\mathtt{lp}'$ be the value of the solution. By Property a, we have $\mathtt{lp}' \le (1 + O(\epsilon))\mathrm{opt}$. Let ψ be integral occurrence-times vector of the solution given by the algorithm. That is, for every $a \in \mathbb{Z}_{\ge 0}$, ψ_a is the number of clients in C with connection distance a. So our goal is to upper bound $\mathbb{E}[\overline{\mathtt{top}\text{-}cn}(\psi)]$ in terms of $\mathtt{lp}'$.

We define a vector $\delta' \in \mathcal{S}_n$ as follows: $\delta'_{a'} = \sum_{p,i,j:\lfloor r_p\rfloor = a'} x_{ij}^{(p)}$ for every $a' \in \mathbb{Z}_{\ge 0}$. That is, for every p, i, j, we include $x_{ij}^{(p)}$ fractional connection of distance $\lfloor r_p \rfloor$ in δ'. By Corollary 4.5 and (8), we have

$$\delta' \preceq_{\frac{1+\epsilon}{1-\epsilon}} \delta.$$

To see the inequality, we consider how δ and δ' is constructed: For every $p \in P$ and $i \in \mathtt{ball}_F(p, r_p)$, we include $x_{ij}^{(p)}$ fraction of distance $d(i,j)$ in δ for every j, and we include $\sum_{j\in C} x_{ij}^{(p)}$ fraction of distance $\lfloor r_p \rfloor \le \frac{1+\epsilon}{1-\epsilon} \sum_{j\in C} d(i,j)x_{ij}^{(p)} / \sum_{j\in C} x_{ij}^{(p)}$ in δ'. Thus Corollary 4.5 can be applied.

The main lemma we prove is the following:

Lemma 4.14. *$\mathbb{E}[\psi]$ is dominated by $\left(1 - \frac{1}{e}\right)\delta + \frac{1}{e}\cdot(\delta \oplus \delta' \otimes 2)$.*

We show why the lemma implies the desired approximation ratio. By the concavity of the $\overline{\mathtt{top}\text{-}cn}(\cdot)$ function in Lemma 4.2, we have

$$\begin{aligned}\mathbb{E}\left[\overline{\mathtt{top}\text{-}cn}(\psi)\right] &\le \overline{\mathtt{top}\text{-}cn}(\mathbb{E}[\psi]) \le \overline{\mathtt{top}\text{-}cn}\left(\left(1-\frac{1}{e}\right)\delta + \frac{1}{e}\cdot(\delta\oplus\delta'\otimes 2)\right)\\ &\le \left(1 + \frac{2}{ec}\cdot\frac{1+\epsilon}{1-\epsilon}\right)\cdot\overline{\mathtt{top}\text{-}cn(\delta)} \le \left(1 + \frac{2}{ec} + O(\epsilon)\right)\mathrm{opt}.\end{aligned}$$

Proof of Lemma 4.14. We focus on a single client $j \in C$. It is optimal to connect j to the nearest open facility. But for the sake of analysis, it is convenient to connect j to a random and possibly sub-optimal facility. Also, the connection cost we impose on j could be larger than its actual cost.

Abusing notations slightly, we use a pair (p, i) to denote the copy of facility i dedicated to $\mathtt{ball}_F(p, r_p)$. The probability that one facility in $\{(p,i) : x_{ij}^{(p)} > 0\}$ is open is at least $1 - \frac{1}{e}$. Using a contention resolution scheme, we can connect j to 0 or 1 open facility in the set, such that

$$\Pr[j \text{ connected to } (p,i)] = \left(1 - \frac{1}{e}\right)x_{ij}^{(p)}, \forall (p,i).$$

When j is connected to (p, i), we impose a connection cost of $d(i, j)$ for j. In this type, we say j is *directly* connected.

With the remaining probability of $1-\left(1-\frac{1}{e}\right)\sum_{i,p} x_{ij}^{(p)} = 1-\left(1-\frac{1}{e}\right) = \frac{1}{e}$, we say j is *indirectly* connected. We then specify the connection cost in this type. For every facility (p, i) with $x_{ij}^{(p)} > 0$, we know that some facility in $\mathtt{ball}_F(p, r_p)$ must be open. Then we can impose a connection cost of $d(i, j) + 2\lfloor r_p \rfloor$ on j by connecting it to the facility in the ball. We make the indirect connection randomly using $x_{ij}^{(p)}$ values:

[5] By allowing a $1 + \epsilon$ loss, we can assume t is an integer power of $1 + \epsilon$, and we only need to enumerate $O(\frac{\log n}{\epsilon})$ different values of t.

conditioned on that we make an indirect connection for j, we randomly choose a pair (p, i) with probabilities $x_{ij}^{(p)}$, and we impose a connection cost of $d(i, j) + 2\lfloor r_p \rfloor$ on j.

Therefore, the connection cost of j is distributed as follows:

- For every facility (p, i), j is directly connected to (p, i) (and thus incurs a cost of $d(i, j)$) with probability $\left(1 - \frac{1}{e}\right) x_{ij}^{(p)}$.
- For every facility (p, i), j is indirectly connected via the facility (p, i) (and thus incurs a cost of $d(i, j) + 2\lfloor r_p \rfloor$) with probability $\frac{1}{e} \cdot x_{ij}^{(p)}$.

Now we consider all clients $j \in C$. We have that $\mathbb{E}[\psi]$ is dominated by $\left(1 - \frac{1}{e}\right)\delta + \frac{1}{e} \cdot \phi$ for some ϕ obtained by adding δ and $\delta' \otimes 2$, which is dominated by $\delta \oplus \delta' \otimes 2$ by Lemma 4.8. Therefore, by Lemma 4.4, we have $\mathbb{E}[\psi]$ is dominated by $\left(1 - \frac{1}{e}\right)\delta + \frac{1}{e} \cdot (\delta \oplus \delta' \otimes 2)$. ☐

4.6 Wrapping up

Therefore, when all steps are successful, and all our guesses are correct, the algorithm returns a solution with expected top-cn norm being at most $\left(1 + \frac{2}{ec} + O(\epsilon)\right)$ opt.

There are $|S \cup R|^k$ different choices for P in Step 4 of Algorithm 4, and $O\left(\frac{\log n}{\epsilon}\right)^k$ different choices for $(r_p)_{p \in P}$ in Step 5. The success probability of Steps 2 and 3 in Algorithm 5 are respectively $1 - \frac{1}{n^2}$ and $\frac{\epsilon^{3k}}{k^k}$. Therefore, if we enumerate all choices of P and $(r_p)_{p \in P}$, and repeat the algorithm $O\left(\frac{k^k}{\epsilon^{3k}} \cdot \log n\right)$ times, the success probability can be increased to $1 - 1/\mathrm{poly}(n)$. Overall, we obtain a final algorithm with running time

$$O\left(\frac{k \log n}{\epsilon^4}\right)^k \cdot \mathrm{poly}(n) \le O\left(\frac{k^2 \log k}{\epsilon^4}\right)^k \cdot \mathrm{poly}(n) \le \mathrm{poly}\left(k, \frac{1}{\epsilon}\right)^k \cdot \mathrm{poly}(n),$$

that with high probability outputs a solution whose top-cn cost is at most $\left(1 + \frac{2}{ec} + O(\epsilon)\right) \cdot$ opt. To see the first inequality, notice that either $\log n \le k \log k$, or $(\log n)^k \le \mathrm{poly}(n)$; thus $(\log n)^k \le \max\{(k \log k)^k, \mathrm{poly}(n)\}$.

References

[1] Karen Aardal, Pieter L.van den Berg, Dion Gijswijt, and Shanfei Li. Approximation algorithms for hard capacitated k-facility location problems. *European Journal of Operational Research*, 242(2):358–368, 2015. 2, 6

[2] Fateme Abbasi, Sandip Banerjee, Jaroslaw Byrka, Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, Dániel Marx, Roohani Sharma, and Joachim Spoerhase. Parameterized approximation schemes for clustering with general norm objectives. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 1377–1399, 2023. 2

[3] Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In *27th Annual European Symposium on Algorithms, ESA 2019*, volume 144, pages 1:1–1:14, 2019. 2, 6

[4] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4), 2020. 2, 6

[5] Soroush Alamdari and David B. Shmoys. A bicriteria approximation algorithm for the k-center and k-median problems. In *Workshop on Approximation and Online Algorithms, WAOA 2017*, volume 10787, pages 66–75, 2017. 1, 3

[6] Hyung-Chan An, Aditya Bhaskara, Chandra Chekuri, Shalmoli Gupta, Vivek Madan, and Ola Svensson. Centrality of trees for capacitated k-center. *Mathematical Programming*, 154(1):29–53, 2015. 1, 2, 3

[7] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, STOC 2001*, pages 21–29, 2001. 5

[8] Sayan Bandyapadhyay, William Lochet, and Saket Saurabh. FPT constant-approximations for capacitated clustering to minimize the sum of cluster radii. In *39th International Symposium on Computational Geometry, SoCG 2023*, volume 258, pages 12:1–12:14, 2023. 6

[9] Judit Barilan, Guy Kortsarz, and David Peleg. How to allocate network centers. *Journal of Algorithms*, 15(3):385–415, 1993. 2

[10] Vladimir Braverman, Vincent Cohen-Addad, Shaofeng H.-C. Jiang, Robert Krauthgamer, Chris Schwiegelshohn, Mads Bech Toftrup, and Xuan Wu. The power of uniform sampling for coresets. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022,*, pages 462–473, 2022. 4

[11] Vladimir Braverman, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for ordered weighted clustering. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 744–753, 2019. 4

[12] Jarosław Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In *Proceedings of the 26h Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, pages 722–736, 2014. 2, 6

[13] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13(2):1–31, 2017. 5

[14] Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13(2):23:1–23:31, 2017. 2

[15] 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, IPCO 2016*, volume 9682, pages 262–274, 2016. 2, 6

[16] Jaroslaw Byrka, Krzysztof Sornat, and Joachim Spoerhase. Constant-factor approximation for ordered k-median. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 620–631, 2018. 2

[17] Deeparnab Chakrabarty and Chaitanya Swamy. Interpolating between k-median and k-center: Approximation algorithms for ordered k-median. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107, pages 29:1–29:14, 2018. 1, 2, 5, 7

[18] Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pages 126–137, 2019. 2

[19] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS 1999*, pages 378–388, 1999. 2, 5

[20] 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. 5

[21] Ke Chen. On k-median clustering in high dimensions. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006*, pages 1177–1185, 2006. 4

[22] Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009. 4

[23] Julia Chuzhoy and Yuval Rabani. Approximating k-median with non-uniform capacities. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005*, volume 5, pages 952–958, 2005. 2, 6

[24] Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2022*, pages 1621–1628, 2022. 2

[25] Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *Journal of the ACM*, 68(6):1–34, 2021. 2

[26] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A $(2+\epsilon)$-approximation algorithm for metric k-median. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 615–624, 2025. 2, 5

[27] Vincent Cohen-Addad, Anupam Gupta, Lunjia Hu, Hoon Oh, and David Saulpic. An improved local search algorithm for k-median. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 1556–1612, 2022. 5

[28] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132, pages 42:1–42:14, 2019. 2, 5

[29] Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132, pages 41–1, 2019. 1, 2, 3, 4, 6

[30] Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. Lp rounding for k-centers with non-uniform hard capacities. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 273–282, 2012. 2

[31] 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*, volume 55, pages 73:1–73:14, 2016. 2, 6

[32] David Eisenstat, Philip N. Klein, and Claire Mathieu. Approximating k-center in planar graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 617–627, 2014. 2

[33] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC 2003*, pages 448–455, 2003. 2

[34] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 21th Annual ACM Symposium on Theory of Computing, STOC 1988*, pages 434–444, 1988. 2

[35] Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the 23rd Annual Symposium on Computational Geometry, SoCG 2007*, pages 11–18, 2007. 2

[36] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020. 4

[37] Qilong Feng, Zhen Zhang, Ziyun Huang, Jinhui Xu, and Jianxin Wang. A unified framework of FPT approximation algorithms for clustering problems. In *31st International Symposium on Algorithms and Computation, ISAAC 2020*, volume 181, pages 5:1–5:17, 2020. 6

[38] Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour. Local search yields a ptas for k-means in doubling metrics. *SIAM Journal on Computing*, 48(2):452–480, 2019. 2

[39] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. 2

[40] Dishant Goyal and Ragesh Jaiswal. Tight FPT approximation for constrained k-center and k-supplier. *Theoretical Computer Science*, 940:190–208, 2023. 2, 3

[41] Dishant Goyal and Ragesh Jaiswal. Tight FPT approximation for socially fair clustering. *Information Processing Letters*, 182:106383, 2023. 6

[42] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999. 2

[43] Anupam Gupta and Kanat Tangwongsan. Simpler analyses of local search algorithms for facility location. *arXiv preprint arXiv:0809.2554*, 2008. 6

[44] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985. 2, 5

[45] Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. 2

[46] Lingxiao Huang, Jian Li, Pinyan Lu, and Xuan Wu. Coresets for constrained clustering: General assignment constraints and improved size bounds. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025*, pages 4732–4782, 2025. 4

[47] Kamal Jain and Vijay V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS 1999*, pages 2–13, 1999. 2

[48] Kamal Jain and Vijay V Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001. 6

[49] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the 18th Annual Symposium on Computational Geometry, SoCG 2002*, pages 10–18, 2002. 6

[50] Samir Khuller and Yoram J Sussmann. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000. 2

[51] Silvio Lattanzi and Christian Sohler. A better k-means++ algorithm via local search. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 3662–3671, 2019. 2

[52] Shanfei Li. An improved approximation algorithm for the hard uniform capacitated k-median problem. *arXiv preprint arXiv:1406.4454*, 2014. 2, 6

[53] Shi Li. Approximating capacitated k-median with $(1+\epsilon)k$ open facilities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 786–796, 2016. 2, 6

[54] Shi Li. On uniform capacitated k-median beyond the natural lp relaxation. *ACM Transactions on Algorithms*, 13(2):1–18, 2017. 2, 6

[55] Shuilian Liu, Xianrun Chen, and Yicheng Xu. A fixed-parameter tractable approximation for capacitated k-supplier. *Theoretical Computer Science*, page 115605, 2025. 6

[56] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 6

[57] Michal Osadnik. Fixed parameter tractable algorithm and coreset for the ordered k-median problem, master thesis. *https://urn.fi/URN:NBN:fi:aalto-202405193486*, 2023. 3

[58] Ján Plesník. A heuristic for the p-center problems in graphs. *Discrete Applied Mathematics*, 17(3):263–268, 1987. 2, 5

[59] Yicheng Xu, Rolf H Möhring, Dachuan Xu, Yong Zhang, and Yifei Zou. A constant parameterized approximation for hard-capacitated k-means. *arXiv preprint arXiv:1901.04628*, 2019. 6

A Making Distances Polynomially Bounded

In this section, we show how to make distances ∞ or integers in $[0, \text{poly}(n)]$, losing only a $(1+\epsilon)$-factor in the approximation ratio, for any symmetric monotone objective norm f. We make a guess on the maximum connection distance L in the optimum solution. Consider the complete bipartite graph between F and C where the lengths of edges are the distances. If an edge has cost more than L, we remove the edge. Otherwise, we round the length down to the nearest integer multiple of $\frac{\epsilon L}{3n^2}$. The new distance is defined as the metric-completion of the graph. If the original distance between two vertices is at most L, then the new distance is in $[L - \frac{2\epsilon L}{3n}, L]$. The optimum cost of the two instances differ by at most the $f(\frac{2\epsilon L}{3n}, \frac{2\epsilon L}{3n}, \cdots, \frac{2\epsilon L}{3n})$, which is at most $\frac{2\epsilon}{3} f(L, 0, 0, \cdots, 0) \leq \frac{2\epsilon}{3} \cdot \text{opt}$. After scaling the distances by $\frac{3n^2}{2\epsilon L}$, all distances are ∞ or integers in $[0, \frac{n^3}{\epsilon}]$.

B Proof of Theorem 2.9

In this section, we prove Theorem 2.9 and Corollary 2.10, which are repeated below for convenience.

Theorem 2.9. *Let $\epsilon > 0$ be a constant. We can efficiently find a set $\mathcal{S}$ of $O\left(\frac{k \ln n}{\epsilon}\right)$ valid stars such that*

- $\biguplus_{(i,J) \in \mathcal{S}} J = C$.
- *For every $j \in C$, let $b = d(i,j)$ for the unique star (i,J) with $j \in J$. Then $h((1-\epsilon)b) \leq \text{opt}'$.*

The algorithm succeeds with high probability.

We solve convex program CP(11), defined as follows.

$$\min \qquad h(d^{\mathsf{av}}) \tag{11}$$

$$\sum_{i \in F} y_i = k \qquad (12)$$

$$x_{ij} \le y_i \qquad \forall i \in F, j \in C \qquad (13)$$

$$\sum_{i \in F} x_{ij} = 1 \qquad \forall j \in C \qquad (14)$$

$$\sum_{j \in C} x_{ij} \le u_i y_i \qquad \forall i \in F \qquad (15)$$

$$d^{\mathsf{av}}_j = \sum_{i \in F} d(i,j) x_{ij} \qquad \forall j \in C \qquad (16)$$

$$x_{ij} \ge 0 \qquad \forall i \in F, j \in C \qquad (17)$$

$$y_i \ge 0 \qquad \forall i \in F \qquad (18)$$

In integer program correspondent to CP(11), y_i indicates whether facility i is open or not, and x_{ij} indicates whether the client j is connected to facility i. (12) restricts us to open k facilities, (13) requires that a client can only be connected to an open facility, (14) requires every client to be connected to a facility, (15) is the capacity constraint, (16) defines the variables d^{av}_j's, and (17) and (18) are the non-negativity constraints.

In the integer programming, we require $x_{ij}, y_i \in \{0,1\}$ for every $i \in F, j \in C$. d^{av}_j is just the connection distance of j. In CP(11) relaxation; we relax the constraint to $x_{ij}, y_i \in [0,1]$. Then d^{av}_j becomes the average connection distance of j over all its fractional connections. The objective is to minimize $h(d^{\mathsf{av}})$, which is convex. So the program is a convex program.

We solve CP (11) to obtain a solution (x, y, d^{av}); let `lp` be the value of the solution. The algorithm is given in Algorithm 7.

Algorithm 7 `Pseudo-Approximation`()

solve CP(11) to obtain $\big((x_{ij})_{i \in F, j \in C}, (y_i)_{i \in F}, (d^{\mathsf{av}}_j)_{j \in C}\big)$
$\mathcal{S} \leftarrow \emptyset$
repeat $\frac{c' k \log n}{\epsilon}$ times for a sufficiently large constant $c' > 0$
    randomly choose a facility $i \in F$ according to the distribution $\{\frac{y_i}{k}\}_{i \in F}$
    choose a set $J \subseteq C$ randomly such that
    - $|J| \le u_i$ with probability 1
    - $\Pr[j \in J] = \frac{x_{ij}}{y_i}$ for every $j \in C$

    ▷ It is well-known that such distribution J exists and we can efficiently generate J according to the distribution.
    add (i, J) to $\mathcal{S}$
if a client j appears in multiple stars in $\mathcal{S}$, we only keep it in the star (i, J) with the smallest $d(i,j)$
return $\mathcal{S}$

Observation B.1. *Each $(i, J) \in \mathcal{S}$ is a valid star.*

Definition B.2. *We say a client j is good if there is a star $(i, J) \in \mathcal{S}$ with $j \in J$ such that $d(i,j) \le \frac{d^{\mathsf{av}}_j}{1-\epsilon}$.*

Lemma B.3. *With probability at least $1 - \frac{1}{n^2}$, all clients in C are good.*

Proof. We say j is covered in an iteration of the repeat loop, if in the iteration, we sampled a star (i, J) with $j \in J$ and $d(i,j) \le \frac{d^{\mathsf{av}}_j}{1-\epsilon}$. Notice that by Markov inequality and that $\sum_{i \in F} y_i = k$, the probability that j is covered in an iteration is at least $\frac{\epsilon}{k}$. The probability that a client j is not covered in any iteration is at most

$$\left(1 - \frac{\epsilon}{k}\right)^{\frac{c' k \log n}{\epsilon}} \le e^{-c' \log n} \le \frac{1}{n^3},$$

when c' is sufficiently large. The lemma follows from the union bound over all clients $j \in C$. $\square$

Theorem 2.9 follows from that h is monotone.

Then we prove Corollary 2.10, which is repeated below:

Corollary 2.10. *Consider the minimum-norm k-clustering problem under a monotone norm f (which is not necessarily symmetric). There is a $(3+\epsilon)$-approximation algorithm for the problem in FPT time with parameters k and ϵ.*

Proof. We let h be the norm f, and the capacities of facilities be ∞. We use Theorem 2.9 to obtain the set $\mathcal{S}$ of valid stars. Let S be the facilities used by $\mathcal{S}$. Then we enumerate all sets $T \subseteq S$ of size k and output the best one.

We then analyze the approximation ratio of the algorithm. Let S^* be the optimum set of k facilities. For every $i^* \in S^*$, we let $\phi(i^*)$ be the facility in S that is nearest to i^*. We claim that $T := \{\phi(i^*) : i^* \in S^*\}$ gives a $(3+\epsilon)$ approximation for the problem. For every $j \in C$, let i^*_j be the facility j is connected to in the solution S^*. Then we have

$$\begin{aligned} d(j,T) &\le d(j,i^*_j) + d(i^*_j,T) \le d(j,i^*_j) + d(i^*_j,\phi(i^*_j)) \le d(j,i^*_j) + d(i^*_j,j) + d(j,\phi(i^*_j)) \\ &\le 2d(j,i^*_j) + d(j,S). \end{aligned}$$

Therefore, the connection distance of j in the solution T is at most twice its connection distance in the optimum solution, plus the connection distance in the solution S. So, its f-norm cost is at most $2\cdot\mathrm{opt} + \frac{\mathrm{opt}}{1-\epsilon} = (3+O(\epsilon))\mathrm{opt}$. $\square$

C Minimum-Norm Capacitated k-Clustering With Given Open Facilities

This section is devoted to the proof of Theorem 2.11, which is repeated below for convenience:

Theorem 2.11. *Consider a minimum-norm capacitated k-clustering instance defined by F, C, n, d, u and k, under a symmetric monotone norm f. Assume $|F| = n = k$ and $\epsilon > 0$ is a constant. We can find a $(1+\epsilon)$-approximate assignment $\sigma \in F^C$ for the instance in time $g(k,\epsilon)\cdot \mathrm{poly}(n)$, for a computable function g depending on k and ϵ.*

C.1 Preliminary

We first provide important definitions useful for this section. Fix $0 < \epsilon < \frac{1}{10}$. For $c \in C$ and $\emptyset \subsetneq X \subseteq F$, say c is X-*exclusive*, if the following two conditions hold:

(i) $\forall f_1, f_2 \in X, \frac{d(f_1,f_2)}{d(c,X)} \le \epsilon$;

(ii) $\forall f \in F\backslash X, \frac{\max_{f'\in X} d(c,f')}{d(c,f)} \le \epsilon$.

If $c \in C$ is not X-exclusive for any $\emptyset \subsetneq X \subseteq F$, say c is *inclusive*. Define C_X to be the set of X-exclusive clients, and C_I as the set of inclusive clients.

We first show that, those X with nonempty C_X forms a laminar family, whose inclusion relation forms a tree structure.

Lemma C.1. *$\{X \mid C_X \ne \emptyset\}$ is a laminar family, i.e. If $X \cap Y \ne \emptyset$, then either $X \subseteq Y$ or $Y \subseteq X$.*

Proof. Suppose the contrary, assuming there exists X and Y satisfying $C_X \ne \emptyset$, $C_Y \ne \emptyset$, $X \cap Y \ne \emptyset$, $X\backslash Y \ne \emptyset$ and $Y\backslash X \ne \emptyset$. Pick $c_X \in C_X$, $c_Y \in C_Y$, $f_X \in X\backslash Y$, $f_Y \in Y\backslash X$ and $f_\cap \in X \cap Y$.

Applied to set X, (i) shows $d(c_X, f_\cap) \ge d(c_X, X) \ge \frac{1}{\epsilon} d(f_X, f_\cap)$, and (ii) further shows $d(c_X, f_Y) \ge \frac{1}{\epsilon}\max_{f'\in X} d(c,f') \ge \frac{1}{\epsilon} d(c_X, f_\cap)$. By triangular inequality, $d(f_Y, f_\cap) \ge d(f_Y, c_X) - d(c_X, f_\cap) \ge \frac{1-\epsilon}{\epsilon} d(c_X, f_\cap) \ge \frac{1-\epsilon}{\epsilon^2} d(f_X, f_\cap)$.

Similarly, $d(f_X, f_\cap) \ge \frac{1-\epsilon}{\epsilon^2} d(f_Y, f_\cap)$, a contradiction as $\frac{1-\epsilon}{\epsilon^2} > 1$ for $\epsilon < \frac{1}{10}$. $\square$

Corollary C.2. $|\{X \mid C_X \neq \emptyset\}| = O(k)$.

In the following lemma, we show that $\{C_X \mid \emptyset \subsetneq X \subseteq F\}$ and C_I partitions C.

Lemma C.3. *For $\emptyset \subsetneq X, Y \subseteq F$, $C_X \cap C_Y \neq \emptyset$ implies $X = Y$.*

Proof. Prove by contradiction. Choose any $c \in C_X \cap C_Y$. Without loss of generality, suppose $X \backslash Y \neq \emptyset$ and choose any $f_1 \in X \backslash Y$ and $f_2 \in Y$. (ii) applied to Y gives $\frac{d(c,f_2)}{d(c,f_1)} \leq \epsilon$. If $f_2 \in Y \backslash X$, (ii) applied to X gives $\frac{d(c,f_1)}{d(c,f_2)} \leq \epsilon$, a contradiction. Otherwise, $f_2 \in X$, (i) applied to X gives $\frac{d(f_1,f_2)}{d(c,f_1)} \leq \epsilon$. According to triangular inequality, $\frac{d(c,f_2)}{d(c,f_1)} \geq 1 - \epsilon$, a contradiction with $\epsilon < \frac{1}{10}$. ☐

Finally, we show that, if a client is invariant, then its distance to any facility is close to the distance between this facility and some other facility. Then, after discretization, there are only $O(k)$ possible distances for each facility.

Lemma C.4. *If $c \in C_I$, then $\forall f_1 \in F, \exists f_2 \in F, \frac{\epsilon}{2} \leq \frac{d(c,f_1)}{d(f_1,f_2)} \leq \frac{2}{\epsilon}$.*

Proof. Suppose the contrary. When $\exists f_1 \in F, \forall f_2 \in F$, either $\frac{d(f_1,f_2)}{d(c,f_1)} < \frac{\epsilon}{2}$ or $\frac{d(c,f_1)}{d(f_1,f_2)} < \frac{2}{\epsilon}$, set $X = \{f_2 \mid \frac{d(f_1,f_2)}{d(c,f_1)} < \frac{\epsilon}{2}\}$. (i) holds by triangular inequality, and (ii) holds trivially. So $c \in C_X$. ☐

Corollary C.5. *For any $f \in F$, $|\{(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} \mid c \in C_I\}| = O(k)$.*

Proof. Lemma C.4 says that, for all $f \in F$, $c \in C_I$, $d(c,f) \in \cup_{f' \in F} \left[\frac{\epsilon}{2} d(f,f'), \frac{2}{\epsilon} d(f,f')\right]$. For each interval, the discretization has $\log_{1+\epsilon} \frac{4}{\epsilon^2} = \frac{1}{\epsilon^{O(1)}} = O(1)$ possible results. Sum over all k intervals, we conclude $|\{d(c,f) \mid c \in C_I\}| = O(k)$. ☐

C.2 Algorithm

We first provide the intuition of the algorithm. Suffer ϵ loss to apply the discretization between distances from C_I to F. Now, consider invariant and exclusive clients separately.

- As Corollary C.5 shows, after discretization, each facility has $O(k)$ possible distances to C_I. As the norm is symmetric, only the number of linkages for each distance matters, so in total $O(k^2)$ integers with value range $[0, n]$ suffice to determine their contribution.
- For X-exclusive clients, (ii) implies that linkages to any facility $f \in F \backslash X$ have distance roughly $d(f, X)$, with ϵ error. Therefore, only the number of linkages matters.
- (i) further implies that, any $c \in C_X$ that links to X has distance approximately $d(c, X)$, with ϵ loss. Since $F \backslash X$ pose no except frequency constraint on the matching from C_X, to minimize the cost from C_X to X, we can always match the nearest clients in C_X to X. Finally, only the number of linkages inside X is useful.

The following lemma formalizes the intuition above.

Lemma C.6. *For solutions $\mathcal{S}$ and $\mathcal{S}'$, $\mathrm{val}(\mathcal{S}') \leq (1 + O(\epsilon))\mathrm{val}(\mathcal{S})$ if $\mathcal{S}'$ satisfies all conditions below:*

- *$\forall f \in F, d \in \{(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} \mid c \in C_I\}$, the number of linkages from C_I to f with discretized distance d is the same between $\mathcal{S}$ and $\mathcal{S}'$;*
- *$\forall \emptyset \subsetneq X \subseteq F, \exists d_0 \in \mathbb{R}$, such that $\forall c \in C_X$, c links to X in $\mathcal{S}'$ if and only if $d(c, X) \leq d_0$.* [6]
- *$\forall \emptyset \subsetneq X \subseteq F, f \in F \backslash X$, the number of linkages from C_X to f is the same between $\mathcal{S}$ and $\mathcal{S}'$;*

[6] Without loss of generality, suppose $d(c, X)$ are distinct for $c \in C_X$.

Proof. Those conditions imply that, we can always find a bijection $\phi : C \to C$, where for all $c \in C$, suppose c matches f in $\mathcal{S}$ and $\phi(c)$ matches f' in $\mathcal{S}'$, then

- If $c \in C_I$, then $\phi(c) \in C_I$, $f' = f$ and $(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} = (1+\epsilon)^{\lceil \log_{1+\epsilon} d(\phi(c),f') \rceil}$;
- If $c \in C_X$ and $f \in X$, then $\phi(c) \in C_X$, $f' \in X$ and $d(\phi(c), X) \le d(c, X)$;
- If $c \in C_X$ and $f \notin X$, then $\phi(c) \in C_X$ and $f' = f$.

To prove the statement of the lemma, it suffices to show $d(\phi(c), f') \le (1+O(\epsilon))d(c,f)$. For the case $c \in C_I$, $(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} = (1+\epsilon)^{\lceil \log_{1+\epsilon} d(\phi(c),f') \rceil}$ directly implies $d(\phi(c), f') \le (1+\epsilon)d(c,f)$. For the case $c \in C_X$ and $f \notin X$, (ii) implies $d(\phi(c), f') \le (1+\epsilon)d(f', X) = (1+\epsilon)d(f,X) \le \frac{1+\epsilon}{1-\epsilon}d(c,f)$. For the case $c \in C_X$ and $f \in X$, (i) implies $d(\phi(c), f') \le \frac{1}{1-\epsilon}d(\phi(c), X) \le \frac{1}{1-\epsilon}d(c,X) \le \frac{1}{1-\epsilon}d(c,f)$. ☐

Lemma C.6 shows that, to construct an $(1+O(\epsilon))$-approximate solution, it suffices to know at most $O(k^2)$ integers in $[0,n]$ about the optimal solution. So the algorithm first guesses such information, and then constructs the solution based on the guess. The algorithm first guesses

- $\forall f \in F, d \in \{(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} \mid c \in C_I\}$, the number of clients in C_I matched to f with discretized distance d, denoted as $n_{f,d}$. Guessing the exact value is unaffordable, so we guess it approximately, meaning that we guess $m_{f,d} \in \{0\} \cup \{(1+\epsilon)^r \mid r \in [0, \lceil \log_{1+\epsilon} n \rceil]\}$, and we say the guess is correct when $\frac{m_{f,d}}{1+\epsilon} \le n_{f,d} \le m_{f,d}$. The number of possible choice for $m_{f,d}$ is $O(\lceil \log_{1+\epsilon} n \rceil) = O(\frac{\ln n}{\epsilon})$.
- $\forall \emptyset \subsetneq X \subseteq F$ and $f \notin X$, the number of clients in C_X matched to f, denoted as $n_{X,f}$. It is also guessed approximately as above.

We use the guessing part to find a solution satisfying all conditions below:

- $\forall f \in F, d \in \{(1+\epsilon)^{\lceil \log_{1+\epsilon} d(c,f) \rceil} \mid c \in C_I\}$, the number of clients matched from C_I to f with discretized distance d is bounded by $m_{f,d}$;
- $\forall \emptyset \subsetneq X \subseteq F$, define $t_X = \max(0, |C_X| - \sum_{f \notin X} m_{c,f})$, then the top-$t$ nearest clients to X in C_X are matched to X. All other clients in C_X are arbitrarily allocated to facilities in $F \backslash X$, such that the set A_f allocated to $f \notin X$ have size at most $m_{X,f}$. Clients in A_f can be matched to f or X.
- All clients are matched to some facility, and the capacity constraint for all facilities is satisfied.

For the second condition, as the number of facilities matched in X is understated, we should allow facilities allocated to facilities outside X to match X to fulfill the capacity constraint. However, measuring their cost as their distance to f is always safe.

As the optimal solution simultaneously satisfies all above conditions with careful allocation in the second condition, there exists a solution. The solution can be easily found by a maximum matching algorithm in polynomial time.

Approximation Ratio When we know all the information exactly, Lemma C.6 shows that the final solution gives $1+O(\epsilon)$ approximation. When guessing approximately, all information is at most $(1+\epsilon)$ times the correct value. From sub-additivity, exceeding part affects the norm by at most ϵ fraction. Finally, the approximation ratio is still $(1+O(\epsilon))$.

Runtime All except the guessing part can be done in polynomial time. For the guessing, we need to guess $O(k^2)$ numbers with $O(\frac{\ln n}{\epsilon})$ choices each, so the total runtime is $(\frac{\ln n}{\epsilon})^{O(k^2)} n^{O(1)}$, which is bounded by $\left(\frac{k}{\epsilon}\right)^{O(k^2)} \cdot \mathrm{poly}(n)$.

D Missing Proofs from Section 4

Lemma 4.2. *For every real $\ell \in [0, n]$, we have that $\overline{\texttt{top-}\ell}(\cdot)$ concave over $\mathcal{S}_n$.*

Proof. Let $\delta, \delta' \in \mathcal{S}_n$, and $\beta \in [0,1]$. Let α and α' be the vectors that define $\overline{\texttt{top-}\ell}(\delta)$ and $\overline{\texttt{top-}\ell}(\delta')$ respectively in Definition 4.1. Then $\beta\alpha + (1-\beta)\alpha'$ is a candidate solution for the LP in the definition of $\overline{\texttt{top-}\ell}(\beta\delta + (1-\beta)\delta')$. Therefore, $\overline{\texttt{top-}\ell}(\beta\delta + (1-\beta)\delta') \geq \overline{L_1}(\beta\delta + (1-\beta)\delta') = \beta \cdot \overline{L_1}(\delta) + (1-\beta)\cdot\overline{L_1}(\delta') = \beta \cdot \overline{\texttt{top-}\ell}(\delta) + (1-\beta)\cdot\overline{\texttt{top-}\ell}(\delta')$. □

Lemma 4.4. *Assume $\delta^1, \delta^2, \cdots, \delta^H, \delta'^1, \delta'^2, \cdots, \delta'^H$ are $2H$ vectors in $\mathcal{S}_n$, $\gamma \in \mathbb{R}_{>0}$, and $\delta'^h \preceq_\gamma \delta^h$ for every $h \in [H]$. Let $\beta_1, \beta_2, \cdots, \beta_H \in [0,1]$ satisfy $\sum_{h=1}^H \beta_h = 1$. Then*

$$\sum_{h=1}^{H} \beta_h {\delta'}^h \preceq_\gamma \sum_{h=1}^{H} \beta_h \delta^h.$$

Proof. Let $\delta = \sum_{h=1}^H \beta_h \delta^h$ and $\delta' = \sum_{h=1}^H \beta_h \delta'^h$. Our goal is to prove $\delta' \preceq_\gamma \delta$. We need to show that $\overline{\texttt{top-}\ell}(\delta') \leq \gamma \cdot \overline{\texttt{top-}\ell}(\delta)$, for any $\ell \in [0, n]$. Fix $\ell \in [0, n]$.

Then, we have some $0 \leq \alpha \leq \delta'$ with $|\alpha'|_1 = \ell$ and $\overline{L_1}(\alpha') = \overline{\texttt{top-}\ell}(\delta')$. Then, there are vectors $\alpha'^0, \alpha'^1, \cdots, \alpha'^H \in \mathbb{R}_{\geq 0}^{\mathbb{Z}_{\geq 0}}$ satisfying

- $\alpha' = \sum_{h=0}^H \beta_h \alpha'^h$, and
- $0 \leq \alpha'^h \leq \delta'^h$ for every $h \in [H]$.

For every $h \in [H]$, define $\ell_h = |\alpha'^h|_1$; then, $\overline{\texttt{top-}\ell_h}(\delta'^h) \geq \overline{L_1}(\alpha'^h)$. There exists some $0 \leq \alpha^h \leq \delta^h$ such that $|\alpha^h|_1 = \ell_h$ and $\overline{\texttt{top-}\ell_h}(\delta^h) = \overline{L_1}(\alpha^h)$. By that $\delta'^h \preceq_\gamma \delta^h$, we have

$$\overline{L_1}(\alpha'^h) \leq \overline{\texttt{top-}\ell_h}(\delta'^h) \leq \gamma \cdot \overline{\texttt{top-}\ell_h}(\delta^h) = \gamma \cdot \overline{L_1}(\alpha^h).$$

Also notice that $\sum_{h=1}^H \beta_h \ell_h = \ell$. Define $\alpha = \sum_{h=1}^H \beta_h \alpha^h$. Then, we have $0 \leq \alpha \leq \delta$, $|\alpha|_1 = \sum_{h=1}^H \beta_h |\alpha^h|_1 = \sum_{h=1}^H \beta_h \ell_h = \ell$, and

$$\begin{aligned} \overline{L_1}(\alpha) &= \overline{L_1}\Big(\sum_{h=1}^H \beta_h \alpha^h\Big) = \sum_{h=1}^H \beta_h \overline{L_1}(\alpha^h) \\ &\geq \frac{1}{\gamma}\sum_{h=1}^H \beta_h \overline{L_1}(\alpha'^h) = \frac{1}{\gamma}\cdot \overline{L_1}\Big(\sum_{h=1}^H \beta_h \alpha'^h\Big) = \frac{1}{\gamma}\cdot\overline{L_1}(\alpha') = \frac{1}{\gamma}\cdot\overline{\texttt{top-}\ell}(\delta'). \end{aligned}$$
□

Corollary 4.5. *Let $\delta \in \mathcal{S}_n$, and $\delta = \sum_{h=0}^H \theta^h$ where $H \in \mathbb{Z}_{\geq 0}$ and $\theta^0, \theta^1, \cdots, \theta^H \geq 0$. Let $\gamma \geq 1$. For every $h \in [H]$, let $b_h \in \mathbb{Z}_{\geq 0}$ satisfy $b_h \leq \gamma \cdot \frac{\overline{L_1}(\theta^h)}{|\theta^h|_1}$. Let $\delta' = \theta^0 + \sum_{h=1}^H |\theta^h|_1 \cdot \mathbf{e}_{b_h}$, where $\mathbf{e}_b$ for any $b \in \mathbb{Z}_{\geq 0}$ is the vector in $\mathbb{R}_{\geq 0}^{\mathbb{Z}_{\geq 0}}$ with $(\mathbf{e}_b)_b = 1$ and $(\mathbf{e}_b)_{b'} = 0$ if $b' \neq b$.*

Then $\delta' \preceq_\gamma \delta$.

Proof. We define δ^h for every integer $h \in [0, H]$ to be $n \cdot \frac{\theta^h}{|\theta^h|_1}$. Define $\delta'^0 = \delta^0$, and $\delta'^h = n \cdot \mathbf{e}_{b_h}$ for every $h \in [H]$. Then $\delta'^h \preceq_\gamma \delta^h$ for every integer $h \in [0, H]$; notice that $\gamma \geq 1$. Define $\beta_h = \frac{|\theta^h|_1}{n}$ for every $h \in [0, H]$. Then,

$$\delta = \sum_{h=0}^H \beta_h \delta^h \qquad \text{and} \qquad \delta' = \sum_{h=0}^H \beta_h \delta'^h.$$

By Lemma 4.4, we have $\delta' \preceq_\gamma \delta$. □

Theorem 4.9. *Let $\delta, \delta' \in \mathcal{S}_n$ such that $\delta' \preceq_\gamma \delta$ for some $\gamma > 0$. Let $c \in (\frac{1}{e}, 1]$ and $\alpha \in [0, c]$. Then we have*

$$\overline{\texttt{top-}cn}\big((1-\alpha)\delta + \alpha \cdot (\delta \oplus \delta' \otimes 2)\big) \le \left(1 + \frac{2\alpha\gamma}{c}\right) \overline{\texttt{top-}cn}(\delta).$$

Proof. There are two reals $x, y \in [0, 1]$ with $x \le c \le y$ such that

$$(1-\alpha)x + \alpha y = c, \quad \text{and}$$
$$\overline{\texttt{top-}cn}\big((1-\alpha)\delta + \alpha \cdot (\delta \oplus \delta' \otimes 2)\big) = (1-\alpha) \cdot \overline{\texttt{top-}xn}(\delta) + \alpha \cdot \overline{\texttt{top-}yn}(\delta \oplus \delta' \otimes 2).$$

We have

$$\overline{\texttt{top-}yn}\big((\delta \oplus \delta' \otimes 2)\big) \le \overline{\texttt{top-}yn}(\delta) + 2 \cdot \overline{\texttt{top-}yn}(\delta') \le (1 + 2\gamma) \cdot \overline{\texttt{top-}yn}(\delta).$$

The second inequality is by that $\delta' \preceq_\gamma \delta$. Therefore,

$$\begin{aligned} \overline{\texttt{top-}cn}\big((1-\alpha)\delta + \alpha(\delta \oplus \delta' \otimes 2)\big) &\le (1-\alpha) \cdot \overline{\texttt{top-}xn}(\delta) + \alpha \cdot \overline{\texttt{top-}yn}(\delta) + 2\alpha\gamma \cdot \overline{\texttt{top-}yn}(\delta) \\ &\le \overline{\texttt{top-}cn}(\delta) + 2\alpha\gamma \cdot \frac{1}{c} \cdot \overline{\texttt{top-}cn}(\delta) \\ &= \left(1 + \frac{2\alpha\gamma}{c}\right) \cdot \overline{\texttt{top-}cn}(\delta). \end{aligned}$$

The second inequality used that $\overline{\texttt{top-}zn}(\delta)$ for a fixed δ is a concave function over $z \in [0, 1]$, and $\overline{\texttt{top-}yn}(\delta) \le \overline{\texttt{top-}n}(\delta) \le \frac{1}{c} \cdot \overline{\texttt{top-}cn}(\delta)$. ☐

E $(3, 1 + \frac{2}{ec} + \epsilon)$-Bi-Criteria Approximation for $(L_\infty, \texttt{top-}cn)$-Norms k-Clustering Problem

In this section, we prove Theorem 1.4 by giving the $(3, 1 + \frac{2}{ec})$-bi-criteria approximation for the $(L_\infty, \texttt{top-}cn)$-norms k-clustering problem, for $c \in (\frac{1}{e}, 1]$.

Definition E.1. *In the $(L_\infty, \texttt{top-}cn)$-Norms k-Clustering problem, we are given F, m, C, n, k and d as in Definition 2.7, and a number $L \in \mathbb{R}_{\ge 0}$, the goal of the problem is to find a set $S \subseteq F$ of k facilities so as minimize $|\vec{d}(S)|_{\texttt{top-}cn}$, subject to $|\vec{d}(S)|_\infty \le L$.*

Let opt *be the optimum value of the instance. A solution $S \subseteq F, |S| = k$ is called an (α, β)-bicriteria approximation for the instance, for some $\alpha, \beta \ge 1$, if $|\vec{d}(S)|_\infty \le \alpha L$ and $|\vec{d}(S)|_{\texttt{top-}cn} \le \beta \cdot \mathrm{opt}$.*

To prove Theorem 1.4, the main modification to Algorithm 5 is that in Step 2, after applying Theorem 2.9 for the $\texttt{top-}cn$ norm to obtain $\mathcal{S}$ and S, we add $O(k \log n)$ facilities that form a 1-approximation (this can be achieved using LP rounding or Theorem 2.9) for the L_∞ norm to S. Now S is a $(1 + O(\epsilon))$-approximation for $\texttt{top-}cn$ norm and has L_∞ cost L. Then, when guessing r_p's in Step 5, we guarantee $r_p \le L$. In the end, we return the solution with the smallest $\texttt{top-}cn$ cost, whose L_∞ cost at most $3L$.

Clearly, if our guesses are correct, then the solution constructed have L_∞ cost at most $3L$. Consider an optimum cluster (i^*, J), where every $j \in J$ has $d(i^*, j) \le L$. If it is of type-1 or 2, the pivot p is in J and thus has $d(i^*, p) \le L$. We are guaranteed to open a facility with distance $r_p \le L$ to p. Therefore, the distance of the facility to i^* is at most $2L$, implying that the distance from all clients J to i^* is at most $3L$. When (i^*, J) is of type-3, then the nearest facility to i^* in S has distance at most $2L$ to i^*, and thus distance at most $3L$ to all clients in J. Finally, for the case $c \in (0, \frac{1}{e}]$, a $(3, 3 + \epsilon)$-approximation can be achieved easily by modifying Algorithm 7 in the proof of Theorem 2.9.