FPT Constant-Approximations for Capacitated Clustering to Minimize the Sum of Cluster Radii

Sayan Bandyapadhyay
Department of Computer Science, Portland State University, OR, USA

William Lochet
LIRMM, Université de Montpellier, CNRS, Montpellier, France

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India

Abstract

Clustering with capacity constraints is a fundamental problem that attracted significant attention throughout the years. In this paper, we give the first FPT constant-factor approximation algorithm for the problem of clustering points in a general metric into k clusters to minimize the sum of cluster radii, subject to non-uniform hard capacity constraints (CAPACITATED SUM OF RADII). In particular, we give a $(15 + \epsilon)$-approximation algorithm that runs in $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$ time.

When capacities are uniform, we obtain the following improved approximation bounds.

- A $(4 + \epsilon)$-approximation with running time $2^{\mathcal{O}(k \log(k/\epsilon))}n^3$, which significantly improves over the FPT 28-approximation of Inamdar and Varadarajan [ESA 2020].
- A $(2 + \epsilon)$-approximation with running time $2^{\mathcal{O}(k/\epsilon^2 \cdot \log(k/\epsilon))}dn^3$ and a $(1+\epsilon)$-approxim- ation with running time $2^{\mathcal{O}(kd \log((k/\epsilon)))}n^3$ in the Euclidean space. Here d is the dimension.
- A $(1 + \epsilon)$-approximation in the Euclidean space with running time $2^{\mathcal{O}(k/\epsilon^2 \cdot \log(k/\epsilon))}dn^3$ if we are allowed to violate the capacities by $(1 + \epsilon)$-factor. We complement this result by showing that there is no $(1 + \epsilon)$-approximation algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, if any capacity violation is not allowed.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Clustering, FPT-approximation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.12

Related Version *Full Version*: https://arxiv.org/abs/2303.07923

Funding *Saket Saurabh*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 819416), and Swarnajayanti Fellowship (no. DST/SJF/MSA01/2017-18).

1 Introduction

The SUM OF RADII (clustering) problem is among the most popular and well-studied clustering models in the literature, together with k-center, k-means, and k-median [12, 25, 4, 30]. In SUM OF RADII , we are given a set P of n points in a metric space with distance dist, and a non-negative integer k specifying the number of clusters sought. We would like to find: (i) a subset C of P containing k points (called centers) and a non-negative integer r_q (called radius) for each $q \in C$, and (ii) a function assigning each point $p \in P$ to a center $q \in C$ such that $\mathsf{dist}(p, q) \leq r_q$. The goal is to minimize the sum of the radii $\sum_{q \in C} r_q$. Alternatively, the objective is to select k balls in the metric space centered at k distinct points of P, such that each point $p \in P$ is contained in at least one of those k balls and the sum of the radii of the balls is minimized.

39th International Symposium on Computational Geometry (SoCG 2023).
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 12; pp. 12:1–12:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In a seminal work, Charikar and Panigrahy [12] studied the Sum of Radii problem. As mentioned in their paper, sum of radii objective can be used as an alternative to the k-center objective to reduce the so called *dissection effect*. The k-center objective is similar to sum of radii, except here one would want to minimize the maximum radius. As in k-center all balls are assumed to have the same maximum radius, the balls can have huge overlap. Consequently, points that should have been assigned to the same cluster might end up in different clusters. This phenomenon is called the dissection effect which can be reduced by using the sum of radii objective instead.

Considering the sum of radii objective, Charikar and Panigrahy [12] obtained a 3.504-approximation running in polynomial time, which is the best known approximation factor for this problem in polynomial time to date. Their algorithm is based on a primal-dual scheme coupled with an application of Lagrangean relaxation. Subsequently, Gibson et al. [23] obtained a $(1+\epsilon)$-approximation in quasi-polynomial time. It follows from the standard complexity theoretic assumptions that the problem cannot be APX-hard. We note that the problem is known to be NP-hard even in weighted planar metrics and metrics of constant doubling dimension [23]. Surprisingly, the problem can be solved in polynomial time in Euclidean spaces when the dimension is fixed [24]. When the dimension is arbitrary, one can obtain a $(1+\epsilon)$-approximation in $2^{\mathcal{O}((k\log k)/\epsilon^2)} \cdot n^{O(1)}$ time, extending the coreset based algorithm for k-center [5].

1.1 Our Problem and Results

In this work, we are interested in the capacitated version of Sum of Radii . Clustering with capacity constraints is a fundamental problem and has attracted significant attention recently [10, 8, 11, 13, 21, 33, 34, 7, 22, 1, 37, 19]. Indeed, capacitated clustering is relevant in many real-life applications, such as load balancing where the representative of each cluster can handle the load of only a bounded number of objects. It is widely known that clustering problems become much harder in the presence of capacity constraints.

Formally, in the Capacitated Sum of Radii problem, along with the points of P in a metric space, we are also given a non-negative integer η_q for each $q \in P$, which denotes the capacity of q. The goal is similar to the goal of Sum of Radii except here each chosen center $q \in C$ can be assigned at most η_q points of P. Alternatively, each cluster contains a bounded number of points specified with respect to the center of the cluster. In the uniform-capacitated version of the problem, $\eta_p = \eta_q$ for all $p, q \in P$, and we denote the capacity by U. We note that in this work, we only consider *hard* capacities, i.e., each point can be chosen at most once to be a cluster center. In this setting, a major open question is to determine whether there is a polynomial time $\mathcal{O}(1)$-approximation algorithm for Capacitated Sum of Radii , even in the uniform-capacitated case.

Question 1: Does Capacitated Sum of Radii admit a polynomial time constant-approximation algorithm, even with uniform capacities?

Designing polynomial time constant-approximations for capacitated clustering problems are notoriously hard. In fact such algorithms exist only for the k-center objective out of the four objectives mentioned before. For uniform capacitated k-center, Khuller and Sussmann [31] designed a 6-approximation improving a 10-approximation of Bar-Ilan et al. [6] who introduced the problem. The first constant-approximation in the non-uniform case [20] was designed after 12 years, which was subsequently improved to a 9-approximation by An et al. [3]. The capacitated problems with k-means and k-median objectives have attracted a lot of attention over the years. But, despite a recent progress for the uniform version in $\mathbb{R}^2$ [14], where a PTAS is achieved, even in $\mathbb{R}^3$, the problem of finding a polynomial time constant-approximation remains open. The best-known polynomial time approximation factor in general metrics is $\mathcal{O}(\log k)$ [19], which is based on a folklore tree embedding scheme.

Technical Barriers for Sum of Radii. The problem with sum of radii objective also appears to be fairly challenging. The main difficulty in achieving a polynomial time $\mathcal{O}(1)$-approximation for Capacitated Sum of Radii is obviously the presence of the capacity bounds even if they are uniform, which makes the problem resilient to the techniques used for solving Sum of Radii . The only polynomial time $\mathcal{O}(1)$-approximation known for Sum of Radii is via a primal-dual scheme. However, it is not clear how to interpret the capacity constraints in the primal, in the realm of dual. Also, while the algorithms for capacitated k-center use LP-relaxation of the natural LP, the standard LP relaxation for Capacitated Sum of Radii has a large integrality gap [29]. Needless to say, the situation becomes much more intractable in the non-uniform capacitated case.

Hardness of Approximation. The lower bounds known for capacitated clustering are equally frustrating as their upper bounds. Surprisingly, the only known lower bounds are the ones for the uncapacitated versions, and hence trivially translated to the capacitated case. Due to the 20-year old work of Guha and Khuller [26], k-median and k-means are known to be NP-hard to approximate within factors of 1.735 and 3.943, respectively. In a recent series of papers, Cohen-Addad, Karthik and Lee [16, 17, 18] have obtained improved constant lower bounds for various clustering problems in different metrics and settings. In particular, in the last work, they introduced an interesting Johnson Coverage Hypothesis [18] which helped them obtain improved bounds in various metrics. As mentioned before, Sum of Radii cannot be APX-hard, and hence there is no known inapproximability results that can be translated to the capacitated version.

In the light of the above discussions, one may conclude that the rather benign capacity constraints have played a bigger role compared to the choice of objective function, in our current lack of understanding of practical clustering models. Therefore, it seems that making any intermediate progress towards understanding capacitated clustering, irrespective of the objective function, is significant and timely.

Coping with Capacitated Clustering. In order to improve the understanding of these challenging open questions, researchers have mainly studied two types of relaxations to obtain constant-approximation algorithms. The more traditional approach taken for k-means and k-median is to design bi-criteria approximation where we are allowed to violate either capacity or the bound on the number of clusters by a small amount [10, 8, 11, 13, 21, 33, 34]. The other (relatively newer) approach is to design *fixed-parameter tractable* (FPT) approximation, thus allowing an extra factor $f(k)$ in the running time. We note that, in recent years, FPT approximations are designed for classic problems improving the best known approximation factors in polynomial time, e.g., k-vertex separator [32], k-cut [27] and k-treewidth deletion [28].

FPT Approximation for Clustering. In the context of clustering problems, the number of clusters k is a natural choice for the parameter, as the value of k is typically small in practice, e.g., $k \leq 10$ in [35, 36]. Consequently, the approach of designing FPT approximation have become fairly successful for clustering problems and have led to interesting results which are not known or impossible in polynomial time. For example, constant-approximations are obtained for the capacitated version of k-median and k-means [1, 37, 19], which almost match the polynomial time constant approximation factors in the uncapacitated case. In the uncapacitated case of k-median and k-means, tight $(1.735 + \epsilon)$ and $(3.943 + \epsilon)$-factor FPT approximations are recently obtained [15, 1, 37], whereas the best known factors in polynomial time are only 2.611 [9] and 6.357 [2]. These results are interesting in particular, as a popular belief in the clustering community is that there is no algorithmic separation between FPT and polynomial time in general metrics (for example, see the comment in [17] after Theorem 1.3). We note that it is possible to obtain $(1 + \epsilon)$-approximations in high-dimensional Euclidean spaces [7, 22], which is impossible in polynomial time, assuming standard complexity theoretic conjectures.

Inamdar and Varadarajan [29] adapted the approach of designing FPT approximation to study the Capacitated Sum of Radii problem with uniform capacities. They make the first substantial progress in understanding this problem through the lens of fixed-parameter tractability. In particular, they obtained a 28-approximation algorithm for this problem that runs in time $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(1)}$. Unfortunately, their algorithm does not work in the presence of non-uniform capacities. Based on their result, the following natural questions arise.

Question 2: Does Capacitated Sum of Radii admit a constant-approximation algorithm, in FPT time, even when capacity constraints are non-uniform?
Question 3: Does Capacitated Sum of Radii admit a $(1 + \epsilon)$-approximation algorithm, in FPT time, when the points are in $\mathbb{R}^d$ (Euclidean Metric)?

We make significant advances towards answering Questions 2 and 3. Our first result completely answers Question 2.

▶ **Theorem 1.** *For any constant $\epsilon > 0$, the Capacitated Sum of Radii problem admits a $(15 + \epsilon)$-approximation algorithm with running time $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$.*

Next, we consider the uniform-capacitated version and prove the following theorem significantly improving over the approximation factor of 28 in [29].

▶ **Theorem 2.** *For any constant $\epsilon > 0$, there exists a randomized algorithm for the Capacitated Sum of Radii problem with uniform capacities that outputs with constant probability a $(4 + \epsilon)$-approximate solution in time $2^{\mathcal{O}(k \log(k/\epsilon))} \cdot n^3$.*

The approximation factor in the above result is interesting in particular, as it almost matches the approximation factor of 3.504 in the uncapacitated case and keeps the avenue of obtaining a matching approximation in polynomial time open.

Finally, we mention the Euclidean version of the problem where we show that adapting the standard coreset argument for regular k-clustering allows us to obtain the following two results.

▶ **Theorem 3.** *For any constant $\epsilon > 0$, there exists a randomized algorithm for the Euclidean version of Capacitated Sum of Radii with uniform capacities that outputs with constant probability a $(2 + \epsilon)$-approximate solution in time $2^{\mathcal{O}((k/\epsilon^2) \log(k/\epsilon))} \cdot dn^3$, where d is the dimension.*

▶ **Theorem 4.** *For any constant $\epsilon > 0$, the Euclidean version of Capacitated Sum of Radii admits an $(1 + \epsilon)$-approximation algorithm with running time $2^{\mathcal{O}(kd \log((k/\epsilon)))} n^3$.*

We also complement our approximability results by hardness bounds. The NP-hardness of Capacitated Sum of Radii trivially follows from the NP-hardness of Sum of Radii . We strengthen this bound by showing the following result.

▶ **Theorem 5.** *Capacitated Sum of Radii with uniform capacities cannot be solved in $f(k)n^{o(k)}$ time for any computable function f, unless ETH is false. Moreover, it does not admit any FPTAS, unless P=NP.*

We also show an inapproximability bound in the Euclidean case even with uniform capacities.

▶ **Theorem 6.** *The Euclidean version of Capacitated Sum of Radii with uniform capacities does not admit any FPTAS even if $k = 2$, unless P=NP.*

Although the above bound does not eradicate the possibility of obtaining a $(1+\epsilon)$-approximation in the Euclidean case, it shows that to obtain such an approximation, even when $k = 2$, one needs $n^{f(\epsilon)}$ time for some non-constant function f that depends on ϵ. This is in contrast to the uncapacitated version of the problem, where one can get $(1+\epsilon)$-approximation in $2^{\mathcal{O}((k \log k)/\epsilon^2)} \cdot n^{O(1)}$ time as mentioned before.

As by products of our techniques we have also obtained improved bi-criteria approximations for the uniform-capacitated version of the problem where we are allowed to use $(1+\epsilon)U$ capacity.

▶ **Theorem 7.** *There is a randomized algorithm for Capacitated Sum of Radii with uniform capacities that runs in time $2^{\mathcal{O}(k \log(k/\epsilon))} \cdot n^{\mathcal{O}(1)}$ and returns a solution with constant probability, such that each ball in the solution uses at most $(1+\epsilon)U$ capacity and the cost of the solution is at most $(2+\epsilon) \cdot \mathrm{OPT}$, where OPT is the cost of any optimal solution in which the balls use at most U capacity.*

The above theorem improves the approximation factor in Theorem 2. In the Euclidean case, we obtain a similar algorithm.

▶ **Theorem 8.** *There is a randomized algorithm for the Euclidean version of Capacitated Sum of Radii with uniform capacities that runs in time $2^{\mathcal{O}((k/\epsilon^2) \log k)} \cdot dn^3$ and returns a solution with constant probability, such that each ball in the solution uses at most $(1+\epsilon)U$ capacity and the cost of the solution is at most $(1+\epsilon) \cdot \mathrm{OPT}$, where OPT is the cost of any optimal solution in which the balls use at most U capacity.*

Note that, by Theorem 6, a result as in the above theorem is not possible if we are not allowed to violate the capacity.

1.2 Preliminaries

Capacitated Sum of Radii . We are given a set P of n points in a metric space with distance dist, a non-negative integer η_q for each $q \in P$, and a non-negative integer k. The goal is to find: (i) a subset C of P containing k points and a non-negative integer r_q for each $q \in C$, and (ii) a function $\mu : P \to C$, such that for each $p \in P$, $\mathsf{dist}(p, \mu(p)) \leq r_{\mu(p)}$, for each $q \in C$, $|\mu^{-1}(q)| \leq \eta_q$, and $\sum_{q \in C} r_q$ is minimized. We will sometimes use OPT to denote the value of an optimal solution.

In the uniform-capacitated case, we denote the common capacity of all centers by U. In the general metric version of our problem, we assume that we are given the pairwise distances dist between the points in P. In the Euclidean version, P is a set of points in $\mathbb{R}^d$ for some $d \geq 1$, dist is the Euclidean distance and any point in $\mathbb{R}^d$ can be selected as a center. Moreover, the capacities of all these centers are uniform.

We denote the ball with center c and radius r by $B(c,r)$. For any ball $B = B(c,r)$, we will use $\mathsf{ext}(B, r')$ to denote the ball $B(c, r+r')$. Sometimes we will also use $\mathsf{rad}(B)$ to denote the radius of B. Let S be a set of points in $\mathbb{R}^d$. The *minimum enclosing ball* of S, noted $\mathsf{MEB}(S)$ is the smallest ball in R^d containing all points of S. We say a ball B *covers* a point p if p is in B.

One important remark regarding solving our capacitated clustering problem on P is that, given a set of k balls $\mathcal{B}$, the problem of deciding whether there is a valid assignment $\mu : P \to \mathcal{B}$ satisfying the capacities can easily be modeled as a bipartite matching problem. This implies in particular that if such an assignment exists, it can also be found in $\mathcal{O}(\sqrt{k}n^{3/2})$ time. Therefore, in all our descriptions of the algorithms, we will focus on finding the solution balls while ensuring that a valid assignment exists.

Another remark is that, in the case where every ball has capacity U, we can assume that $|P| \leq k \cdot U$, or the instance is a trivial NO instance.

Organization. Due to limited place, we chose to only present the proof of Theorem 1. The rest of the proofs can be found in the extended version.

2 Capacitated Sum of Radii : General Metric

In this section, we study the case of non uniform capacities. In this setting, for every point x of P, there is an associated integer η_x and any ball centered at x can be assigned at most η_x points. The uniform case correspond to the case where $\eta_x = U$ for all $x \in P$. For convenience, we restate the theorem statement.

▶ **Theorem 1.** *For any constant $\epsilon > 0$, the* Capacitated Sum of Radii *problem admits a $(15+\epsilon)$-approximation algorithm with running time $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$.*

From now on, let $\mathcal{B}^\star := \{B_1^\star, \cdots, B_k^\star\}$ denote the set of balls of a hypothetical optimal solution, $\mu^\star : P \to \mathcal{B}^\star$ be the associated assignment and for all $i \in [k]$, let $r_i^\star$ and $c_i^\star$ denote the radius and the center of the ball $B_i^\star$, respectively. By Lemma 9, just below, we can assume that the algorithm knows an approximate radius r_i for each $r_i^\star$. For a ball $B_i^\star \in \mathcal{B}^\star$, we say that a ball B_i is an *approximate* ball of $B_i^\star$ if $B_i^\star \subseteq B_i$, and if x_i denotes the center of B_i, then $\eta_{x_i} \geq \eta_{c_i^\star}$. Note that because of the capacity constraints, we can associate $(\mu^\star)^{-1}(B_i^\star)$ to B_i.

Let us first mention that it is possible to guess an approximation to each of the radii, r_i, in polynomial time. The proof can be found in the extended version.

▶ **Lemma 9.** *For every $0 < \epsilon < 1$, there exists a randomized algorithm, running in linear time that finds with probability at least $\frac{\epsilon^k}{k^k \cdot n^2}$ a set of reals $\{r_1, \ldots, r_k\}$ such that for every $i \in [k]$, $r_i^\star \leq r_i$ and $\sum_{i \in [k]} r_i \leq (1+\epsilon) \sum_{i \in [k]} r_i^\star$.*

From now on we *assume for simplicity that the algorithm knows an approximate value r_i of $r_i^\star$ for all $i \in [k]$*. Let us give some informal ideas about how the algorithm of Theorem 1 works. Some technicalities, especially about making sure we don't pick the same center twice, will be left out to the more formal description of the algorithm.

Informal sketch

Ideally we would like to find for each optimal ball $B_i^\star$ an approximate ball B_i having the same center as $B_i^\star$ and a radius $r_i \leq C \cdot r_i^\star$, for some constant C. Indeed, if we have such a set of balls, then the obvious assignment μ defined as $\mu(x) = B_i$ whenever $\mu^\star(x) = B_i^\star$ would give a solution. While this is not possible in general, the algorithm will start with a greedy procedure to get a set of approximate balls $\mathcal{B}_1$ for some indices I_1. The procedure is quite simple: start with $\mathcal{B}_1 := \emptyset$, $I_1 = \emptyset$ and as long as the union of balls in $\mathcal{B}_1$ does not cover P, pick a point x of P not in the union, guess the index i such that $\mu^\star(x) = B_i^\star$ and pick c the point at distance at most r_i of x which maximises the value of η_c. Since $c_i^\star$ is at distance at most r_i of x, we have that $\eta_{c_i^\star} \leq \eta_c$ and that $\mathsf{dist}(c, c_i^\star) \leq 2r_i$, which means that the ball B_i of radius $5r_i$ centered around c is an approximate ball of $B_i^\star$. Therefore, the algorithm will add B_i to $\mathcal{B}_1$ and the index i to I_1. This procedure stops when the union of $\mathcal{B}_1$ covers P. At the end of that first step, we have that $\mathcal{B}_1$ contains an approximate ball for each ball $B_i^\star$ of radius $5r_i$ with $i \in I_1$. And while the union of $\mathcal{B}_1$ covers all P, we are far from being done as the capacity constraints have not been taken into account.

Consider now a ball $B_j^\star$ such that $j \notin I_1$, which means that no approximate ball of $B_j^\star$ is in $\mathcal{B}_1$. In the best case (Lemma 13 below), there is a ball $B_i \in \mathcal{B}_1$ of center x_i approximating $B_i^\star$ such that $5r_i \leq r_j$ and $B_i \cap B_j^\star$ is non empty. Indeed, in that case the ball of radius $5r_i + r_j$ around x_i contains $c_j^\star$, the center of $B_j^\star$. This means that if x is the point in that ball maximizing η_x, then the ball of center x and radius $2 \cdot (5r_i + 2r_j) \leq 4r_j$ contains $x_j^\star$ and thus the ball B_j of center x and radius $2 \cdot (5r_i + 2r_j) + r_j \leq 5r_j$ contains $B_j^\star$ and is an approximate ball of $B_j^\star$. Therefore, if such indices j and i exist, the algorithm can guess them and add a new approximate ball to $\mathcal{B}_1$.

After this second step, we reach a point where, if $j \notin I_1$ and $i \in I_1$ are such that $B_j^\star \cap B_i \neq \emptyset$, then $r_j \leq 5r_i$. In particular, incurring an extra $5r_i$ as we just did to get a replacement for $xc_j^\star$ is too costly. For this reason, at this step of the algorithm we stop trying to find approximate balls and instead focus on finding balls to "fix" the assignment. Since the balls in $\mathcal{B}_1$ are approximate balls, it means that we can replace $B_i^\star$ with $B_i \in \mathcal{B}_1$ for any $i \in I_1$ (and take the other $B_j^\star$), and still have a solution to our problem with slightly bigger balls. Abusing notation we can still use $\mu^\star$ for the valid assignment. Now for an index $j \notin I_1$, the ball $B_j^\star$ intersects a subset, say T_j, of balls in $\mathcal{B}_1$. Ideally we would like to find a ball B_j of center x and radius r_j such that $\eta_x \geq \eta_{x_j}$ and $|B_j \cap (\mu^\star)^{-1}(B_i)| \geq |(\mu^\star)^{-1}(B_j^\star) \cap B_i|$ for all $i \in T_j$. Indeed, in that case we could replace $B_j^\star$ by B_j and the condition on $B_j \cap B_i^\star$ ensures that we could adapt the assignment $\mu^\star$ to be a valid assignment by assigning $(\mu^\star)^{-1}(B_j^\star) \cap B_i$ to B_i and a set of the same size in $B_j \cap (\mu^\star)^{-1}(B_i)$ to B_j.

The main difficulty here is that even if we guess the set T_j, picking B_j greedily is not possible as there might be some competitions between the sizes of the intersection with the different balls in B_i for $i \in T_j$ (we cannot afford to guess the $|(\mu^\star)^{-1}(B_j^\star) \cap B_i|$). The way to avoid this problem is to expand all the balls B_i of $\mathcal{B}_1$ by $10r_i$. Indeed, since we have assumed that $r_j \leq 5r_i$ for every $i \in T_j$, it means now that $B_j^\star$ is **entirely contained** in the $\mathsf{ext}(B_i, 10r_i)$ (expanded ball) for $i \in T_j$. So now, denoting P_j to be the intersection of all the $\mathsf{ext}(B_i, 10r_i)$ for $i \in T_j$ where we removed all the other $B_{i'}$ for $i' \in I_1 \setminus T_j$, we have that $(\mu^\star)^{-1}(B_j^\star)$ is a subset of P_j, and we can take B_j as the ball of center x and radius r_j which maximizes $s_x = \min\{\eta_x, |B(x, r_j) \cap P_j|\}$. Here there are some technicalities if B_j intersects some ball $B_{j'}^\star$ for $j' \notin I_1$ (that includes $j = j'$), but assume for the moment that it is not the case and let us hint why we can actually replace $B_j^\star$ by B_j in our solution if all the balls $B_i \in \mathcal{B}_1$ are replaced by $\mathsf{ext}(B_i, 10r_i)$. Indeed, by choice of B_j, we can assign s_x points of P_j to B_j. By our assumptions, all these points were assigned to B_i for $i \in T_j$ in $\mu^\star$, which means that by assigning these points to B_j, there is now a new budget s_x of available points in the union of $\mathsf{ext}(B_i, 10r_i)$ for $i \in T_j$. However, since $B_j^\star$ is entirely contained in the $\mathsf{ext}(B_i, 10r_i)$ for $i \in T_j$, and by choice of s_x, we can assign all the elements of $(\mu^\star)^{-1}(B_j^\star)$ to the balls $\mathsf{ext}(B_i, 10r_i)$ for $i \in T_j$ using this new budget.

Therefore, the last phase of the algorithm consists in building a set of "replacement" balls $\mathcal{B}_2$ for the balls $B_j^\star$ with $j \notin I_1$ by guessing T_j and building the intersection P_j to take greedily a ball of radius r_j inside that set (see Lemma 14). This is done sequentially, and the set I_2 will contain all indices j for which $\mathcal{B}_2$ contains a replacement ball B_j for $B_j^\star$. An important remark here is that the properties required for balls in $\mathcal{B}_2$ are dependent on the balls in $\mathcal{B}_1$ and not just the optimal balls. For technical reasons, we might have to add a new ball in $\mathcal{B}_1$ during the process of building $\mathcal{B}_2$, in which case we cannot guarentee that the properties of balls in $\mathcal{B}_2$ hold anymore. If this happens, the algorithm will then erase all the choices of $\mathcal{B}_2$ and I_2 and start the second phase again. However, since we only do this when I_1 gets bigger, this is done at most k times before $I_1 = [k]$. The algorithm stops when the sets I_1 and I_2 contains all indices of $[k]$ which means that each ball $B_j^\star$ either has an approximate ball in $\mathcal{B}_1$ or a replacement ball in $\mathcal{B}_2$.

The algorithm

As explained previously, the algorithm maintains two disjoint sets of indices I_1 and I_2, initially set to $\emptyset$, as well as two sets of balls $\mathcal{B}_1$ and $\mathcal{B}_2$, also initially set to $\emptyset$. $\mathcal{B}_1$ and $\mathcal{B}_2$ will eventually contain a representative ball B_i for every ball $B_i^\star$ in the optimal solution. Moreover, we will argue that there exists a valid assignment of the points to the balls (with an expansion) in the union of these two sets.

For every $i \in I_2$, let T_i denote the set of indices j of I_1, such that $B_i^\star \cap B_j$ is not empty, and P_i denote the intersection of the extensions $\mathsf{ext}(B_j, 10r_j)$ over all $j \in T_i$ after removing the points of B_s for $s \in I_1 \setminus T_i$.

We say that the sets $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ form a *valid configuration* if the following properties are satisfied.

- For every $i \in I_1$, there is an approximate ball $B_i \in \mathcal{B}_1$ of $B_i^\star$ of radius at most $5r_i$.
- For every $i \in I_2$, T_i is non-empty, $B_i^\star \subseteq P_i$ and there exists a ball $B_i \in \mathcal{B}_2$ of center x_i and radius r_i, such that both η_{x_i} and $B_i \cap P_i$ have size at least $|(\mu^\star)^{-1}(B_i^\star)|$.
- For $i, j \in I_2$, B_i and B_j do not intersect.
- For every $i \in I_2$ and $s \notin I_1$, $B_s^\star$ and B_i do not intersect.
- For every $j \in [k]$, if $c_j^\star$ is a center of a ball in $\mathcal{B}_1$ (respectively, $\mathcal{B}_2$), then $j \in I_1$ (respectively, I_2) and $c_j^\star$ is the center of B_j.

Before describing the algorithm to construct a valid configuration $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ such that $I_1 \cup I_2 = [k]$, let us show that such a configuration would indeed yield an approximate solution.

▶ **Lemma 10.** *Let $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ be a valid configuration such that $I_1 \cup I_2 = [k]$, then the set of balls $\mathcal{B}$ containing the balls in $\mathcal{B}_2$, as well as for every $i \in I_1$ the ball $\mathsf{ext}(B_i, 10r_i)$ is a 15-approximate solution.*

Proof. The fact that the sum of radii of the balls in $\mathcal{B}$ is at most 15 times the optimal solution follows from the definition. To prove the lemma, we have to show that this is a valid solution by giving a valid assignment. Recall that $\mu^\star$ is the assignment for $\mathcal{B}^\star$.

For every $i \in I_2$, recall that T_i denotes the set of indices j of I_1 such that $B_i^\star$ intersects B_j and P_i denotes the intersection of all the $\mathsf{ext}(B_j, 10r_j)$ for $j \in T_i$ where we removed the points in B_s for $s \in I_1 \setminus T_i$ for $j \in I_1$. By definition of a valid configuration, if we use Y_i to denote the set $(\mu^\star)^{-1}(B_i^\star)$, then there exists a set X_i of size $|Y_i|$ in $B_i \cap P_i$. The following claim is a crucial part of the proof.

▷ Claim 11. Any point $x \in X_i$ is such that $\mu^\star(x) = B_j^\star$ for some $j \in T_i$

Proof. Indeed, by definition, the only balls $B_j \in \mathcal{B}_1$ containing an element x of P_i are such that $j \in T_i$, so in particular if $j \in I_1 \setminus T_i$, $B_j^\star \subseteq B_j$ doesn't contain x. Moreover, if $j \in I_2$, then we know by definition that $B_j^\star \cap B_i$ is empty (that includes $B_i^\star$). Since $x \in B_i$, this concludes the proof of our claim. ◁

The previous claim implies that, if we define for every $j \in T_i$ the set $X_{i,j} := X_i \cap (\mu^\star)^{-1}(B_j^\star)$, then the $X_{i,j}$ for $j \in T_i$ actually defines a partition of X_i. As $|X_i| = |Y_i|$, we can partition the set Y_i into sets $Y_{i,j}$ for $j \in T_i$ such that $|X_{i,j}| = |Y_{i,j}|$ for all $j \in T_i$. Remember that, for $j \in T_i$, $Y_i \subseteq \mathsf{ext}(B_j, 10r_j)$. By convention, if $j \in T_1 \setminus T_i$, then $X_{i,j}$ and $Y_{i,j}$ are defined as the empty set.

For every $j \in I_1$, we can now define $L_j = \big((\mu^\star)^{-1}(B_j^\star) \setminus (\cup_{i \in I_2} X_{i,j})\big) \cup_{i \in I_2} Y_{i,j}$. Since the sets B_i, for $i \in I_2$, are pair-wise disjoint and $|X_{i,j}| = |Y_{i,j}|$ for all elements $j \in I_1$ and $i \in I_2$, we have that $|L_j| = |(\mu^\star)^{-1}(B_j^\star)|$. Moreover, since $Y_{i,j}$ is non empty only if $Y_i \subseteq \mathsf{ext}(B_j, 10r_j)$, it means that $L_j \subseteq \mathsf{ext}(B_j, 10r_j)$ and because B_j is an approximate ball of $B_j^\star$, it means that the center x_j of B_j is such that $\eta_{x_j} \geq |(\mu^\star)^{-1}(B_j^\star)| = |L_j|$.

Finally this means that the function μ such that $\mu^{-1}(\mathsf{ext}(B_j, 10r_j)) = L_j$ for all $j \in I_1$ and $\mu^{-1}(B_i) = X_i$ for all $i \in I_2$ is a valid assignment from P to $\mathcal{B}$, which ends the proof. ◀

Now, we describe the algorithm that constructs the desired configuration. The first phase of the algorithm will consist of a greedy selection of elements in I_1, such that the union of $\mathcal{B}_1$ covers P (Lemma 12). As said previously, this will not imply that we can assign points to these balls, without violating capacity constraints. The following two other lemmas (Lemma 13 and 14) will be used to achieve that.

As we deal with hard capacities, we cannot reuse any center. We need the following definition to enforce that. We call a point $p \in P$ an *available center*, if p has not already been selected as a center of a ball in $\mathcal{B}_1$ or $\mathcal{B}_2$.

▶ **Lemma 12.** *If $(I_1, I_2 = \emptyset, \mathcal{B}_1, \mathcal{B}_2 = \emptyset)$ is a valid configuration such that the union of the balls in $\mathcal{B}_1$ do not cover P, then there exists a randomized algorithm, running in linear time and with probability at least $1/2k^2$, that finds an index s and a ball B_s such that adding s to I_1 and B_s to $\mathcal{B}$ still yields a valid configuration.*

Proof. Let x be any point in P not covered by the union of the balls in $\mathcal{B}_1$ and i be the index such that $\mu^\star(x) = B_i^\star$. Let c be the available potential center in P at distance at most r_i from x which maximises the value of η_c. If c is not a center of some $B_j^\star$ for $j \notin (I_1 \cup I_2)$, then the ball B_i of center c and radius $3r_i$ is an approximate ball of $B_i^\star$ and thus adding B_i to $\mathcal{B}_1$ and i to I_1 yields a valid configuration. If c is a center of some $B_j^\star$ for $j \notin (I_1 \cup I_2)$, then adding the ball B_j of center c and radius r_j to $\mathcal{B}_1$ and j to I_1 also yields a valid configuration.

The algorithm will then pick uniformly at random an index $i' \in [k]$, then decide uniformly at random wether the available center c' at distance at most $r_{i'}$ is a center of some $B_j^\star$ for some $j \notin (I_1 \cup I_2)$. If it decides negatively, then the algorithm will output $s := i'$ and B_s the ball of center c' and radius $3r_{i'}$. If it decides positively, then the algorithm will then also pick uniformly at random and index $j' \in [k]$ and output $s := j'$ as well as B_s the ball of center c' and radius $r_{j'}$.

The algorithm then suceeds if $i' = i$, if it decides correctly if c' is a center of some $B_j^\star$ and if $j' = j$ in the case where it is. Overall this is true with probability at least $\frac{1}{k \cdot 2 \cdot k}$, which ends the proof. ◀

The first phase of the algorithm consists of applying the algorithm from Lemma 12 until the union of the balls in $\mathcal{B}_1$ covers all the points in P. The next two lemmas are used in the next phase of the algorithm.

▶ **Lemma 13.** *If $(I_1, I_2 = \emptyset, \mathcal{B}_1, \mathcal{B}_2 = \emptyset)$ is a valid configuration such that the balls in $\mathcal{B}_1$ cover the points of P and there exist two indices $i \in I_1$ and $j \in [k] \setminus (I_1 \cup I_2)$ such that B_i and $B^\star_j$ intersect and $r_j \geq 5r_i$, then there exists a randomized algorithm that in linear time and with probability at least $1/2k$, finds an index $t \in [k] \setminus (I_1 \cup I_2)$ and a ball B_t such that $(I_1 \cup \{t\}, I_2, \mathcal{B}_1 \cup \{B_t\}, \mathcal{B}_2)$ is a valid configuration.*

Proof. Let x_i denote the center of B_i, and B' be the ball of center x_i and radius $5r_i + r_j$. Because B_i is an approximate ball of $B^\star_i$, and $B^\star_j$ and B_i intersect, we have that B' contains $c^\star_j$. Let x be the potential center of B' which maximises η_x. If there exists an index $j' \in [k] \setminus (I_1 \cup I_2)$, such that the ball $B^\star_{j'}$ is centered at x, then $t := j'$ and the ball B_t of center x and radius $r_{j'}$ satisfy the property of the lemma (remember that $\mathcal{B}_2 = \emptyset$). If not, then the ball B_j at center x and of radius $2 \cdot (5r_i + r_j) + r_j$ is an approximate ball of $B^\star_j$ of radius at most $5r_j$. Indeed, because the ball at center x_i and of radius $(5r_i + r_j)$ contains both x and $c^\star_j$, it means that the ball at center x and of radius $2(5r_i + r_j)$ contains $c^\star_j$ and thus $B^\star_j \subseteq B_j$. Again, as $\mathcal{B}_2 = \emptyset$, then $t := j$ and $B_t := B_j$ satisfy the properties of the lemma.

The algorithm therefore consists of choosing uniformly at random which of the two cases is true. In the first case it also chooses uniformly at random an index $j_1 \in [k]$ and outputs $t := j_1$ as well as the ball B_t of center x and radius r_{j_1}. In the second case it outputs $t := j$ as well as the ball B_t of center x and radius $2 \cdot (5r_i + r_j) + r_j$. The previous discussion implies that the algorithm succeeds if it chooses correctly between the two cases, and in the first case if $j_1 = j'$. Overall, the probability of success is at least $1/2k$. ◀

▶ **Lemma 14.** *Suppose $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ is a valid configuration with the property that the balls in $\mathcal{B}_1$ cover the points of P and for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and $B^\star_i$ intersect, $r_i \leq 5r_j$. Then there exists a randomized algorithm that in linear time and with probability at least $1/4k^2$, either finds an index $t \in [k] \setminus (I_1 \cup I_2)$ and a ball B_t such that $(I_1 \cup \{t\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_t\}, \mathcal{B}_2 = \emptyset)$ is a valid configuration, or finds an index $s \in [k] \setminus (I_1 \cup I_2)$ and a ball B_s such that $(I_1, I_2 \cup \{s\}, \mathcal{B}_1, \mathcal{B}_2 \cup \{B_s\})$ is a valid configuration.*

Proof. Let i be the element of $[k] \setminus (I_1 \cup I_2)$ minimizing r_i, and let T_i denote the set of indices $j \in I_1$ such that $B_j \cap B^\star_i$ is non-empty. By the hypothesis of the lemma, we have that $r_i \leq 5r_j$ for each element $j \in T_i$. In particular, it means that $B^\star_i \subseteq \mathsf{ext}(B_j, 10r_j)$ for every $j \in T_i$. Let P_i denotes the intersection of all those sets $\mathsf{ext}(B_j, 10r_j)$ where we removed B_s for all $s \in I_1 \setminus T_i$. Let x be the available center in P_i such that, denoting B_x the ball at center x and of radius r_i, B_x is disjoint from all the elements in $\mathcal{B}_2$ and $s_x = \min\{\eta_x, |B_x \cap P_i|\}$ is maximized. Note that because $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ is a valid configuration, $x^\star_i$ is an available center in P_i and $B^\star_i$ does not intersect any of the balls in $\mathcal{B}_2$. This implies that $s_x \geq |(\mu^\star)^{-1}(B^\star_i)|$. If B_x does not intersect any ball $B^\star_{i'}$ with $i' \in [k] \setminus (I_1 \cup I_2)$, then by the above discussion we have that setting $B_i = B_x$, $(I_1, I_2 \cup \{i\}, \mathcal{B}_1, \mathcal{B}_2 \cup \{B_i\})$ is a valid configuration.

Suppose now that B_x intersects some $B^\star_{i'}$ with $i' \in [k] \setminus (I_1 \cup I_2)$. ($i'$ can also be i.) In that case, the ball at center x and of radius $r_i + r_{i'}$ contains $c^\star_{i'}$. Since $r_i \leq r_{i'}$ by the choice of i, this means that the ball $B_{i'}$ of center x and radius $r_i + 2r_{i'} \leq 3r_{i'}$ is an approximate ball of $B^\star_{i'}$. There might be several options for i', but we can just make an arbitrary choice. The only thing to be careful about is if $x = c^\star_{i''}$ for some $i'' \in [k] \setminus (I_1 \cup I_2)$, then the algorithm will pick that index and add the ball $B_{i''}$ of center x and radius $r_{i''}$ to $\mathcal{B}_1$. In any case, $(I_1 \cup \{i'\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_{i'}\}, \mathcal{B}_2 = \emptyset)$ or $(I_1 \cup \{i''\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_{i''}\}, \mathcal{B}_2 = \emptyset)$, depending on that last condition, is a valid configuration, in particular as $\mathcal{B}_2 = \emptyset$.

Finally, the algorithm just decides between these two cases randomly, and in the second case picks uniformly the index i' and then outputs the described ball and index. In the second case, it also needs to decide if there exists i'' such that $x = c^\star_{i''}$ and in which case pick that index uniformly at random as well. Overall, the probability of success of this algorithm is at least $1/4k^2$. ◀

We are now ready to prove our main theorem.

Proof of Theorem 1. Let us describe the algorithm. First, it applies Lemma 9 to obtain an approximation r_i of each $r^\star_i$ with probability at least $\frac{\epsilon^k}{k^k \cdot n^2}$. Then the algorithm initialize a valid configuration $(I_1 = \emptyset, I_2 = \emptyset, \mathcal{B}_1 = \emptyset, \mathcal{B}_2 = \emptyset)$ and run the algorithm of Lemma 12 at most k times, until $\mathcal{B}_1$ covers all the points of P. At each step, the probability of success is at least $1/k^2$, so in total at least $1/k^{2k}$. Then the algorithm enters into the second phase. This phase is divided into multiple steps.

In the beginning of each step, we maintain the invariant that the current configuration is valid with $I_2 = \emptyset$ and $\mathcal{B}_2 = \emptyset$. Each step then consists of a series of applications of Lemma 13 followed by a series of applications of Lemma 14. The current step ends when an index is added to I_1 by the application of Lemma 14, and hence at that point $I_2 = \emptyset$ and $\mathcal{B}_2 = \emptyset$, or $I_1 \cup I_2 = [k]$. We go to the next step (maintaining the invariant), unless $I_1 \cup I_2 = [k]$ in which case the algorithm terminates.

In a step, the algorithm decides which lemma to apply as long as $I_2 = \emptyset$. Otherwise, it applies only Lemma 14. If $I_2 = \emptyset$, it randomly decides if there exists indices $i \in I_1$ and $j \in [k] \setminus (I_1 \cup I_2)$ such that $B^\star_i$ and $B^\star_j$ intersect and $r_j \geq 5r_i$. In which case, the algorithm applies Lemma 13 to increase the size of $|I_1|$ in linear time and with probability at least $1/2^2k$. If no such pair of indices exists, then the algorithm applies Lemma 14 to increase $|I_1 \cup I_2|$ or $|I_1|$ in linear time and with probability at least $1/2k^2$.

▷ Claim 15. Assuming the algorithm made all the correct random choices, it terminates with a valid configuration $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ such that $I_1 \cup I_2 = [k]$ after $O(k^2)$ applications of Lemma 13 and 14.

Proof. First, we argue about the maximum number of applications of the two lemmas. Note that in each step, we add at least one index to I_1 and then only go to the next step. Also, once an index is added to I_1 it is never removed. Thus, the total number of steps is at most k. Also, in each step, everytime we apply a lemma, the size of $I_1 \cup I_2$ gets increased, which can be at most k. Thus, in each step we will apply the lemmas at most k times in total. Hence, the total number of applications of both lemmas is $O(k^2)$.

Next, we prove that the algorithm terminates with the desired configuration. Fix a step. Note that if $I_2 = \emptyset$ and we make correct choices, we can correctly apply a lemma and make progress. This is true, as the conditions in the two lemmas are complementary. Now, if $I_2 \neq \emptyset$, then we have applied Lemma 14 at least once. This implies for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and $B^\star_i$ intersect, $r_i \leq 5r_j$. Hence, the condition of Lemma 13 is false for the current set I_1. Now, we do not change I_1 throughout a step once we apply Lemma 14, except at the last time, in which case we go to the next step emptying I_2. Thus, once $I_2 \neq \emptyset$, throughout the step, it holds that for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and $B^\star_i$ intersect, $r_i \leq 5r_j$. Hence, we can always apply Lemma 14 and make progress.

Now, consider the last step, we prove that at the end of this step $I_1 \cup I_2 = [k]$. By the above discussion, this step ends either if $I_1 \cup I_2$ becomes $[k]$ or an index is added to I_1. In the latter case, we go to the next step. However, as the current step is the last one, it must be the case that $I_1 \cup I_2 = [k]$. This completes the proof of the claim, as we always maintain a valid configuration. ◁

If the algorithm made all the correct random choices, by the above claim together with Lemma 10, we can get a 15-approximate solution from $\mathcal{B}_1$ and $\mathcal{B}_2$. The algorithm runs in linear time and succeeds with probability at least $\frac{\epsilon^k}{k^k \cdot n^2} \cdot \frac{1}{k^{2k}} \cdot \frac{1}{(2k^3)^{k^2}}$. This means that running the previous algorithm $2^{\mathcal{O}(k^2 \log k)} \cdot n^2$ times, we obtain a $(15+\epsilon)$-approximation algorithm with constant probability. Lastly, it is not hard to derandomize this algorithm by performing exhaustive searches in each step instead of making decisions randomly. The time bound still remains the same. ◀

3 Conclusions

In this paper, considering the Capacitated Sum of Radii problem, we obtained the first constant-factor $(15+\epsilon)$ approximation algorithm that runs in FPT time, making significant progress towards understanding the barriers of capacitated clustering. While our techniques are tailor-made for FPT type results, we hope some of the ideas will also be useful in obtaining a similar approximation in polynomial time. We leave this as an open question.

Question 1: Does Capacitated Sum of Radii admit a polynomial time constant-approximation algorithm, even with uniform capacities?

For the problem with uniform capacities, we obtained improved approximation bounds of $4+\epsilon$ and $2+\epsilon$ in general and Euclidean metric spaces, respectively. We also showed hardness bounds in both general and Euclidean metric spaces complementing our approximation results. The following two natural open questions are left by our work.

Question 2: What is the best constant-factor approximation possible for Capacitated Sum of Radii or uniform Capacitated Sum of Radii in FPT time?

Question 3: Does Euclidean Capacitated Sum of Radii admit an $(1+\epsilon)$-approximation algorithm, in $f(k,\epsilon) \cdot n^{g(\epsilon)}$ time for some functions f and g?

References

1 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 1:1–1:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.ESA.2019.1`.

2 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):FOCS17–97, 2019.

3 Hyung-Chan An, Aditya Bhaskara, Chandra Chekuri, Shalmoli Gupta, Vivek Madan, and Ola Svensson. Centrality of trees for capacitated k-center. *Math. Program.*, 154(1-2):29–53, 2015. `doi:10.1007/s10107-014-0857-y`.

4 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

5 Mihai Badoui, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002,*

Montréal, Québec, Canada, STOC '02, pages 250–257, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.509947.

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

7 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k-means problem. *Theory Comput. Syst.*, 62(1):93–115, 2018. doi:10.1007/s00224-017-9820-7.

8 Jaroslaw Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 722–736. SIAM, 2015.

9 Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

10 Jaroslaw Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k-median problem with 1+\epsilon capacity violation. In Quentin Louveaux and Martin Skutella, editors, *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings*, volume 9682 of *Lecture Notes in Computer Science*, pages 262–274. Springer, 2016.

11 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.

12 Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, 68(2):417–441, 2004. doi:10.1016/j.jcss.2003.07.014.

13 Julia Chuzhoy and Yuval Rabani. Approximating k-median with non-uniform capacities. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 952–958. SIAM, 2005.

14 Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2241–2259. SIAM, 2020. doi:10.1137/1.9781611975994.138.

15 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight fpt approximations for k-median and k-means. *arXiv preprint arXiv:1904.12334*, 2019.

16 Vincent Cohen-Addad and CS Karthik. Inapproximability of clustering in lp metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 519–539. IEEE, 2019.

17 Vincent Cohen-Addad, CS Karthik, and Euiwoong Lee. On approximability of clustering problems without candidate centers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2635–2648. SIAM, 2021.

18 Vincent Cohen-Addad and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k-means and k-median in ?p-metrics. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1493–1530. SIAM, 2022.

19 Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.41.

20 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, New Brunswick, NJ, USA, October 20-23, 2012*, pages 273–282. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.63.

21 H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with (1+epsilon)-capacity violation. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016.

22 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020. doi:10.1007/s00453-019-00616-2.

23 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On metric clustering to minimize the sum of radii. *Algorithmica*, 57(3):484–498, 2010. doi:10.1007/s00453-009-9282-7.

24 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On clustering to minimize the sum of radii. *SIAM J. Comput.*, 41(1):47–60, 2012. doi:10.1137/100798144.

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

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

27 Anupam Gupta, Euiwoong Lee, and Jason Li. An fpt algorithm beating 2-approximation for k-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2821–2837. SIAM, 2018.

28 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michał Włodarczyk. Losing treewidth by separating subsets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1731–1749. SIAM, 2019.

29 Tanmay Inamdar and Kasturi R. Varadarajan. Capacitated sum-of-radii clustering: An FPT approximation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 62:1–62:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.62.

30 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. *Comput. Geom.*, 28(2-3):89–112, 2004.

31 Samir Khuller and Yoram J. Sussmann. The capacitated K-center problem. *SIAM J. Discret. Math.*, 13(3):403–418, 2000. doi:10.1137/S0895480197329776.

32 Euiwoong Lee. Partitioning a graph into small pieces with applications to path transversal. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1546–1558. SIAM, 2017.

33 Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015.

34 Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. *ACM Trans. Algorithms*, 13(2):22:1–22:18, 2017.

35 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

36 Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, pages 525–526, 2000.

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