Capacitated Sum-Of-Radii Clustering: An FPT Approximation

Tanmay Inamdar
Department of Computer Science, University of Iowa, Iowa City, IA, USA
tanmay-inamdar@uiowa.edu

Kasturi Varadarajan
Department of Computer Science, University of Iowa, Iowa City, IA, USA
kasturi-varadarajan@uiowa.edu

Abstract

In sum of radii clustering, the input consists of a finite set of points in a metric space. The problem asks to place a set of k balls centered at a subset of the points such that every point is covered by some ball, and the objective is to minimize the sum of radii of these balls. In the capacitated version of the problem, we want to assign each point to a ball containing it, such that no ball is assigned more than U points, where U denotes the *capacity* of the points. While constant approximations are known for the uncapacitated version of the problem, there is no work on the capacitated version. We make progress on this problem by obtaining a constant approximation using a Fixed Parameter Tractable (FPT) algorithm. In particular, the running time of the algorithm is of the form $2^{O(k^2)} \cdot n^{O(1)}$. As a warm-up for this result, we also give a constant approximation for the uncapacitated sum of radii clustering problem with matroid constraints, thus obtaining the first FPT approximation for this problem.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Sum-of-radii Clustering, Capacitated Clustering

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.62

Funding This research is partially supported by the National Science Foundation under Grant CCF-1615845.

1 Introduction

Clustering problems have received a great deal of attention in theoretical as well as practical research. Different ways of modeling a clustering problem have been proposed. A common way to model clustering problems is to assume that the data is represented as a set of points in a finite metric space, and the distance between a pair of points is a measure of similarity between the corresponding data points. Now, we want to partition the set of input points, such that the points belonging to each group are more similar to each other than the points outside the group. In the following, we focus on a particular set of three related clustering objective functions – k-center, k-median, and sum of radii clustering. We first describe the general setup.

Let P be a set of n input points in a metric space, and let d be the corresponding distance function. Let k denote the number of desired clusters, where k is a parameter of the problem. We want to find a set $C \subseteq P$ of at most k centers, such that a certain clustering objective function $\sigma(C, P)$ is minimized. In the k-center problem, the objective function is the largest distance of a point to its nearest center; whereas in the k-median problem, it is the sum of all such distances.

28th Annual European Symposium on Algorithms (ESA 2020).
Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 62; pp. 62:1–62:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Now we describe the closely related sum of radii objective. For any $c \in P$, and $\rho \geq 0$, let $B(c, \rho) = \{p \in P : d(c, p) \leq \rho\}$ denote the ball of radius ρ centered at c. In the sum of radii objective, we want to additionally compute a *radius assignment* $r : C \to \mathbb{R}^+$, such that the corresponding set of balls $\mathcal{B} = \{B(c, r(c)) : c \in C\}$ *covers* the entire set of points P. The objective is to minimize the sum of radii of the balls, i.e., $\sum_{c \in C} r(c)$.

The sum of radii problem was studied by Charikar and Panigrahy [8], who gave a constant approximation. This result is obtained by first obtaining a constant approximation via the primal-dual technique to a closely related problem, which is the Lagrangian relaxation of the original sum of radii problem. Subsequently, Behsaz and Salavatipour [4] improved the approximation ratio in a restricted setting, and Gibson et al. [14] gave a $(1+\epsilon)$-approximation in quasi-polynomial time. In light of the latter result, the problem is likely not APX-hard, under standard complexity theoretic assumptions. There has also been significant work on certain generalizations of this problem. More general objective functions, such as the sum of α'th powers of the radii for a fixed $\alpha \geq 1$, and more general constraints, such as multi-covering of points, have been addressed, but these generalizations are outside the scope of this article.

Constant approximations are also known for the metric k-center and k-median problems, and unlike the sum of radii clustering problem, these problems are known to be APX-hard – see [15, 17, 5] and the references therein for these results. We now focus on a particular generalization of clustering problems which is the focus of this work.

1.1 Capacitated Clustering

A commonly considered generalization of clustering problems is the capacitated clustering, which models the situation where a center is able to provide a certain service to a specific number of points, which are sometimes referred to as *clients* in this context. In the uniform capacitated clustering problem, we are given an integer $1 \leq U \leq n$, which represents the *capacity* of any chosen center. Now, we also want to assign each point to a chosen center, such that no center is assigned more than U points. Here, we also require that, if a point p is assigned to a center c, then it is also covered by the ball placed at c. In a generalization called the non-uniform version of the problems, different centers may have different capacities.

The capacitated sum of radii problem has not yet been considered in the literature. Known techniques do not seem to extend to the capacitated sum of radii problem. Firstly, it can be easily shown that the standard Linear Programming (LP) relaxation has large integrality gap. Furthermore, it is not clear whether it is possible to strengthen this LP by imposing additional constraints implied by the problem structure, as done in [2] for capacitated facility location. Another piece of evidence is a hardness result from [3] for a closely related problem, which can be modified to rule out an $o(\log n)$-approximation in polynomial time, if we want to minimize the sum of α-th powers of radii, where $\alpha > 1$. While this hardness result does not rule out an $O(1)$ approximation in polynomial time for $\alpha = 1$, it does tell us that such a result would need to exploit rather special properties that hold in the $\alpha = 1$ case.

Before describing our results, we review the work on the capacitated versions of the k-center and k-median problems, and look at possible approaches that have been successful for these problems.

Related Work

The capacitated versions of the k-center and k-median problems can be defined analogously. These problems have received a great attention from the researchers. Constant approximations for the capacitated k-center are known [18], even for the non-uniform version. On the other hand, obtaining a constant approximation for the capacitated k-median problem has been a long-standing open problem in approximation algorithms. Researchers have explored different approaches for tackling this problem, in particular, relaxing some of the requirements of traditional approximation algorithms. One such relaxation allows for bi-criteria approximations – constant approximations have been obtained by violating the capacities by a small factor ([7, 10, 6, 13]), or by violating the k-constraint (the number of medians) by a constant factor ([20, 21]).

Yet another recent approach relaxes the requirement that the algorithm run in polynomial time. An algorithm with the running time of $f(p) \cdot n^{O(1)}$ is known as a Fixed Parameter Tractable (FPT) algorithm, where p is a parameter of the problem. Note that, although the running time may depend exponentially (or worse) on the parameter p, the dependence on n, the input size, is strictly polynomial. In the context of capacitated clustering, such an FPT algorithm parameterized by k, the number of clusters, may be acceptable, if k is a small constant. Adamczyk et al. [1] give a $(7+\epsilon)$-approximation for the (uniform/non-uniform) capacitated k-median problem in $k^{O(k)} \cdot n^{O(1)}$ time. They use an approximate solution for the *uncapacitated* k-median problem to convert the given instance into a simpler instance that has more structure, at an expense of a small constant factor loss in the approximation guarantee. Then, they obtain a near-optimal solution for this simpler instance in FPT time. Cohen-Addad and Li [11] improved the approximation guarantee to $3+\epsilon$, again using a similar FPT running time. Their algorithm is based on a coreset construction, and they obtain a constant approximation for this smaller coreset in FPT time. They also obtain an $(1+\epsilon)$-approximation in Euclidean metrics. These results are complemented by Adamczyk et al. [1], who observe that one cannot hope to obtain an exact FPT algorithm even for the *uncapacitated* k-median problem. Parameterized algorithms and complexity is a large and active domain of research, and we direct the reader to a textbook such as [12] for a more detailed background.

1.2 Our Results and Techniques

Our main result is a 28-approximation for the uniform capacitated sum of radii problem, that runs in $2^{O(k^2)} \cdot n^{O(1)}$ time. This result is in a similar vein as the aforementioned results ([1, 11]) for the capacitated k-median problem. Adapting techniques they develop for capacitated k-median, we can obtain an FPT approximation for capacitated sum of radii in metrics of constant doubling dimension. However, for general metrics, we have not been able to adapt their approach. Therefore, we develop a novel algorithm, which we discuss at a high level below.

Fix an optimal solution to the problem. First we discretize the optimal solution by rounding the radii up to a power of $1+\epsilon$ for a fixed $\epsilon > 0$, and now suppose that this discretized solution exactly k_i balls of radius r_i, where each r_i is a power of $(1+\epsilon)$. We show that this first step can be implemented in FPT time. Therefore, we can assume henceforth that we know the "radius profile" of the optimal solution. Our main algorithm proceeds in multiple *levels*. Roughly speaking, the goal of the algorithm at a particular level i is to guess the approximate locations of the k_i optimal balls of radius r_i. However, the size of the search space for this guessing is too large to ultimately obtain an FPT algorithm.

Therefore, we employ a certain *greedy* strategy to guess some balls not chosen in the optimal solution. This allows us to bound the size of the search space, while simultaneously allowing us to argue that an appropriate capacity reassignment is possible if our algorithm misses the approximate location of an optimal ball.

Sum of Radii with a Matroid Constraint

Although the high-level idea of our algorithm is relatively simple, the technical arguments to establish that such a capacity reassignment is possible are quite sophisticated and involved. Therefore, we first consider a related, but simpler, problem as a warm-up. The natural candidate is the uncapacitated version (for which constant approximations are known in polynomial time [8]), but we consider a more general version, which replaces the k-constraint by a more general matroid constraint.

A matroid $\mathcal{M}$, on the set of given points P, is the pair $(P, \mathcal{I})$, where $\mathcal{I}$ is a collection of subsets of P with the following properties: (i) $A \in \mathcal{I}$ implies that $\forall B \subseteq A$, $B \in \mathcal{I}$, (ii) If $A, B \in \mathcal{I}$ with $|B| < |A|$, then there exists a $p \in A \setminus B$, such that $B \cup \{p\} \in \mathcal{I}$. If a set C belongs to $\mathcal{I}$, then C is said to be *independent* in the corresponding matroid $\mathcal{M}$. One consequence of this definition is that all inclusion-wise maximal independent sets of a matroid $\mathcal{M}$ have equal size, and they are called the bases of $\mathcal{M}$.

In the Matroid Sum of Radii problem, a feasible solution consists of a set of centers C, and a radius assignment $r : C \to \mathbb{R}^+$ such that the resulting set of balls covers P. Furthermore, we also require that C be an independent set according to a given matroid $\mathcal{M}$. The objective of the problem is to minimize the sum of radii. We assume that we have an oracle access to an algorithm $\mathcal{A}_\mathcal{M}$ that answers in polynomial time, whether a candidate set of centers is independent in the matroid $\mathcal{M}$. For many "natural" matroids, the definition of an independent set is simple, and thus the oracle can be simulated in a straightforward manner.

We give a $(9 + \epsilon)$-approximation for this problem in $b^{O(b)} \cdot n^{O(1)}$ time, where b is the size of a basis of the matroid $\mathcal{M}$. At a high level, our strategy is similar to our main result for the capacitated sum of radii problem. However, unlike the capacitated version, here we can always find the approximate locations of the optimal centers, which simplifies the algorithm and its analysis. Although this result is not the main contribution of our work, it provides a good vantage point to understand our result for capacitated sum-of-radii clustering.

We note that constant approximations are known for the matroid versions of k-center and k-median [9, 19]. These problems were originally motivated from the so-called red-blue median problem [16], where the centers come in one of the two *types*: red and blue, and we are required to select at most k_r red centers and k_b blue centers to minimize the k-median objective. The matroid formalization captures this scenario as well as its generalization for arbitrary number of types. In particular, the special case of one color corresponds to the k-constraint in the uncapacitated setting.

2 Sum of Radii with a Matroid Constraint

In this problem, we are given a finite metric space (P, d), and a matroid $\mathcal{M} = (P, \mathcal{I})$ on the set of points P. We want to place a set of balls to cover the points in P, while minimizing the sum of radii of the balls. Furthermore, it is required that the set of centers is an independent set in the given matroid.

Formally, we want to find a set of centers that forms an independent set, i.e., $C \in \mathcal{I}$ and assign radii $r : C \to \mathbb{R}^+$, such that $P \subseteq \bigcup_{c \in C} B(c, r(c))$. The objective is to minimize $\sum_{c \in C} r_c$, over all such feasible solutions.

Fix an optimal solution (C^*, r^*), and let $k = |C^*|$. Note that C^* is an independent set in $\mathcal{M}$. First, we guess the value of k by iteratively trying $k' = 1, 2, \ldots, b$, and returning the solution with smallest cost. Here, b denotes the size of any base in $\mathcal{M}$. For a particular value k', the algorithm runs in $k'^{O(k')} \cdot n^{O(1)}$ time. Note that the overall running time of this algorithm is $\sum_{k'=1}^{b} k'^{O(k')} \cdot n^{O(1)} = b^{O(b)} n^{O(1)}$. From now on, we will focus on the iteration where $k' = k$.

Before discussing our main algorithm, we discuss K-CENTER(U, r), which is an important subroutine. Here, $U \subseteq P$ is a set of points to be covered, and $r \geq 0$ denotes the target radius. It is a simple iterative procedure that selects an as-yet uncovered point p, and marks all points in its $2r$-neighborhood as covered. It also adds p to the set of centers Q, and this iterative procedure continues until all points in U are marked as covered. This is a well-known 2-approximation algorithm for the k-center problem, and is summarized in the following lemma.

▶ **Lemma 1.** *Let $U \subseteq P$ be a subset of points, and suppose there exists a set $C \subseteq P$ of k centers such that $\max_{u \in U} d(u, C) \leq r$. Then, the set of centers Q returned by* K-CENTER(U, r') *for any value $r' \geq r$ satisfies: (i) $|Q| \leq k$, and (ii) $\max_{u \in U} d(u, Q) \leq 2r'$.*

Algorithm 1 K-CENTER(U, r).

Initially, all points in U are marked as uncovered, $Q \leftarrow \emptyset$
while there exists an uncovered point in U **do**
    Let $p \in U$ be an uncovered point, add p to Q
    Mark all points in $B(p, 2r)$ as covered
return Q

Preprocessing

First, we discretize the possible choices of radii in the following way. Let $\epsilon > 0$ be a constant, and let R denote the smallest power of $1 + \epsilon$ larger than the maximum radius of any optimal ball – note that we can "guess" the value of R in polynomial time. Furthermore, for any ball with radius smaller than $\frac{\epsilon R}{k}$, we round its radius up to $\frac{\epsilon R}{k}$, and the total increase over at most k such balls is at most ϵR, which is at most ϵ times the optimal cost. Now, we round up radii of all balls to the next larger power of $(1 + \epsilon)$. Note that the resulting solution is within a factor of $(1 + \epsilon)^2$ from the cost of the optimal solution. Furthermore, there are $t = \log_{1+\epsilon} \frac{R}{\frac{R\epsilon}{k}} = O(\frac{1}{\epsilon} \log \frac{k}{\epsilon})$ distinct values of radii. Since ϵ is fixed, $t = O(\log k)$. From now onwards, we will slightly abuse the terminology, and use the terms "optimal solution", "optimal ball" etc. to refer to the corresponding entities in the optimal solution modified in this manner.

Define $r_1 = R, r_2 = \frac{R}{1+\epsilon}, \ldots, r_t = \frac{R}{(1+\epsilon)^{t-1}}$, where $r_{t+1} < \frac{R\epsilon}{k} \leq r_t$. Suppose for every $1 \leq i \leq t$, the optimal solution uses exactly $0 \leq k_i \leq k$ balls of radius r_i. Note that $\sum_{i=1}^{t} k_i = k$. Let $\mathcal{O} = \bigcup_{i=1}^{t} \mathcal{O}_i$ be the set of balls in the optimal solution, where $\mathcal{O}_i \subseteq \mathcal{O}_i$ is the subset of balls of radius r_i. Let $C^* \subseteq P$ denote the set of centers, and let $C_i^* \subseteq C^*$ denote the set of centers of the balls in $\mathcal{O}_i$. For a particular value of i, we define $\mathcal{O}_{<i} = \bigcup_{j=1}^{i-1} \mathcal{O}_j$, and the subsets $\mathcal{O}_{\leq i}, \mathcal{O}_{>i}, \mathcal{O}_{\geq i}$ (resp. $C_{\leq i}^*$ etc.) are defined similarly.

First, we guess the "radius profile" of the optimal solution. There are $O(\log k)$ classes of radii, and for each class r_i, $0 \leq k_i \leq k$. Therefore, the number of overall choices for the radius profile can be upper bounded by $k^{O(\log k)} \ll k^{O(k)}$.

Algorithm 2 MatroidSoR$(\mathcal{B}, i)$.

▷ $1 \le i \le t+1$ is the current level – we want to guess at most k_i centers for balls of radius $4r_i$
▷ $\mathcal{B}$ is the set of balls fixed at earlier levels 1 through $i-1$

$U_i \leftarrow P \setminus \left(\bigcup_{B \in \mathcal{B}} B\right)$ ▷ *Set of points not covered by balls in* $\mathcal{B}$
if $i = t+1$ and $U_i = \emptyset$ **then** ▷ *All points are covered at level* $\le t$
  $\mathcal{D} \leftarrow$ Disjointify$(\mathcal{B})$ ▷ *Procedure* Disjointify *is described before Observation 3 in text*
  for every non-empty subset $\mathcal{D}' \subseteq \mathcal{D}$ **do**
    if balls returned by MatroidIndependentSet$(\mathcal{D}', \mathcal{D})$ cover all points **then**
      output MatroidIndependentSet$(\mathcal{D}', \mathcal{D})$ and **halt**
else if $i = t+1$ and $U_i \neq \emptyset$ **then** ▷ *Not all points are covered by balls at level* $\le t$
  return ▷ $\mathcal{B}$ *is a wrong guess*
$P_i \leftarrow$ K-Center(U_i, r_i) ▷ P_i *is the potential set of centers at level* i
if $|P_i| > k$ **then** ▷ U_i *cannot be covered by at most* k *balls of radius* r_i
  return ▷ $\mathcal{B}$ *is a wrong guess*
else
  For every $C_i \subseteq P_i$ of size at most k_i, call MatroidSoR$(\mathcal{B} \cup \mathcal{B}(C_i), i+1)$
  ▷ $\mathcal{B}(C_i) := \{B(c, 4r_i) : c \in C_i\}$

Algorithm

Having guessed the radius profile $(k_1, k_2, \ldots, k_t)$, our algorithm invokes MatroidSoR$(\emptyset, 1)$ (see Algorithm 2). The procedure MatroidSoR$(\mathcal{B}, i)$ is recursive, and proceeds in multiple *levels*. Fix $1 \le i \le t$, which denotes the current level. We are given a set of balls $\mathcal{B}$ selected at *higher* levels, i.e., levels 1 through $i-1$. For $1 \le j \le i-1$, we let C_j denote the set of centers of balls in $\mathcal{B}$ of level j. We know that $|C_j| \le k_j$, and each $c \in C_j$ has a ball of radius $4r_j$ around it. Now, we want to find a set of at most k_i centers to place balls of radius $4r_i$ at this level.

Let us now see how the algorithm MatroidSoR$(\mathcal{B}, i)$ places these balls at level i. We find U_i, the set of points not covered by any ball in $\mathcal{B}$. We then use algorithm K-Center(U_i, r_i) to find a solution to cover the points in U_i using balls of radius $2r_i$. We will later prove in Lemma 2 that if the set of balls $\mathcal{B}$ added to the solution so far is "correct" (formalized in the Lemma), then the solution P_i returned by the K-center algorithm contains at most k centers. Therefore, if $|P_i| > k$, we conclude that the set of balls $\mathcal{B}$ added to the solution so far is incorrect, and we stop.

Now, suppose $|P_i| \le k$. Then, we enumerate every subset $C_i \subseteq P_i$ of size at most k_i, and recurse on each subset. Note that the number of subsets can be upper bounded by $\sum_{i=0}^{k_i} \binom{k}{k_i} \le k^{O(k_i)}$. Assuming the set $\mathcal{B}$ is "correct", one of these $k^{O(k_i)}$ recursive calls is also "correct". Now we formalize this notion in the following Lemma.

▶ **Lemma 2.** *At any level* $1 \le i \le t$*, in one of the recursive calls to* MatroidSoR*$(\mathcal{B}, i)$, for any optimal center* $c_j^* \in C_j^*$ *with* $1 \le j < i$*, one of the following holds:*

1. *There exists* $c \in C_\ell$ *with* $\ell \le j$*, and* $B(c_j^*, r_j) \subseteq B(c, 4r_\ell)$*, OR*
2. $B(c_j^*, r_j)$ *is completely covered by balls in* $\mathcal{B}$ *of level* 1 *through* $j-1$*. In this case, there exists a center* $c \in C_\ell$ *with* $\ell < j$*, such that* $d(c_j^*, c) \le 4r_\ell$*.*

Proof. We prove this claim inductively.

Base case. This corresponds to $i = 2$. We want to show that, there exists a set $\mathcal{B}$ of balls chosen at level 1, such that, at the start of the algorithm MatroidSoR$(\mathcal{B}, 2)$, every ball $B(c_1^*, r_1) \in \mathcal{O}_1$ is contained in some ball in $\mathcal{B}$.

For the base case, consider the situation at the start of the algorithm, after we invoke MatroidSoR($\emptyset$, 1). Note that $U_1 = P$. Note that the optimal solution covers P using k balls of radius at most r_1. Consider the set P_1 of points returned by K-Center(U_1, r_1). Using Lemma 1, we have that $|P_1| \leq k$, and for any $c_1^* \in C_1^*$, there is some $\varphi(c_1^*) := c \in P_1$ with $d(c_1^*, c) \leq 2r_1$. Let $C_1 = \{\varphi(c_1^*) \mid c_1^* \in C_1^*\}$. Clearly, $|C_1| \leq |C_1^*| = k_1$, and for any $c_1^* \in C_1^*$, $B(c_1^*, r_1) \subseteq B(\varphi(c_1^*), 4r_1)$. Thus, the recursive call MatroidSoR($\mathcal{B}(C_1)$, 2) satisfies the required properties.

Inductive hypothesis. Now we assume that the claim holds inductively at the start of iteration i, and prove that it also holds at level $i+1$ in one of the recursive calls. That is, fix a recursive call MatroidSoR($\mathcal{B}, i$), where $\mathcal{B}$ is a set of balls chosen at levels 1 through $i-1$, such that, any ball $B^* \in \mathcal{O}_{<i}$ is covered by a ball in $\mathcal{B}$, as guaranteed by the induction hypothesis. Now, let U_i^* be the set of points not covered by any such optimal ball (from $\mathcal{O}_{<i}$). Note that inductive hypothesis implies that $U_i \subseteq U_i^*$, which implies that U_i can be covered using at most k balls of radius at most r_i. Now, Lemma 1 implies that the set P_i of points returned by K-Center(U_i, r_i) has size at most k.

We will define a mapping $\varphi : C_i^* \to P_i \cup \{\perp\}$ that specifies a center in P_i whose ball covers $B(c_i^*, r_i)$, if any. Now, consider an optimal center $c_i^* \in C_i^*$, that has a ball $B^* = B(c_i^*, r_i)$ centered at it. We consider two different cases.

Case 1. If there exists a point $p \in B^* \cap U_i$, then using Lemma 1, there exists a center $c \in P_i$ returned by K-Center(U_i, r_i), such that $d(c, p) \leq 2r_i$. Since $d(c_i^*, c) \leq d(c_i^*, p) + d(p, c) \leq 3r_i$, $B(c_i^*, r_i) \subseteq B(c, 4r_i)$. In this case, we define $\varphi(c_i^*) = c$.

Case 2. Otherwise, $B^* \cap U_i = \emptyset$, which implies that all points in B^* are covered by the balls in $\mathcal{B}$ of levels 1 through $i-1$. In particular c_i^* is also covered by a ball $B(c, 4r_\ell)$, where $\ell < i$. In this case, we set $\varphi(c_i^*) = \perp$.

Note that the two cases correspond to the two criteria in the statement of the lemma. Furthermore, if $\varphi(c_i^*) = \perp$, then $B(c_i^*, r_i)$ is covered by one or more balls in $\mathcal{B}$ of levels 1 through $i-1$, i.e., we do not require a ball at level i to cover this ball. Otherwise, $\varphi(c_i^*) \in P_i$, and $B(c_i^*, r_i) \subseteq B(\varphi(c_i^*), 4r_i)$. Let $C_i := \{c \in P_i : \varphi^{-1}(c) \neq \emptyset\}$. Since $|C_i^*| = k_i$, $|C_i| \leq k_i$, and the recursive call corresponding to MatroidSoR($\mathcal{B} \cup \mathcal{B}(C_i), i+1$) satisfies the required properties, recalling that $\mathcal{B}(C_i) := \{B(c, 4r_i) : c \in C_i\}$. ◀

Now, let us discuss the algorithm at level $i = t+1$. Note that Lemma 2 implies that, all points must be covered at level $t+1$ in one of the recursive calls. Therefore, if $U_{t+1} \neq \emptyset$, then we conclude that the set of balls $\mathcal{B}$ is incorrect.

Henceforth, let $\mathcal{B}$ denote the set of balls at level $t+1$ guaranteed by Lemma 2, and focus on the call MatroidSoR($\mathcal{B}, t+1$). Note that $\mathcal{B}$ covers all points, which implies that U_{t+1} is empty. We now call the procedure Disjointify($\mathcal{B}$), which we describe now. In this procedure, we assign each point in P to the largest ball in $\mathcal{B}$ that covers it, breaking ties arbitrarily. Let $D_j(c)$ denote the set of points assigned to a particular ball $B(c, 4r_j) \in \mathcal{B}$. Note that $D_j(c) \subseteq B(c, 4r_j)$; however the inclusion may be strict. In particular, it may be the case that $c \notin D_j(c)$, or $D_j(c)$ may even be empty.

Let $\mathcal{D}$ be the resulting collection of sets in $\mathcal{B}$ that are made disjoint in this manner. In order to distinguish the resulting disjoint sets from the original set of balls, we refer to them as *clusters*. The following observations follow from the definition of $\mathcal{B}$ and the description of Disjointify.

▶ **Observation 3.**

1. *The clusters in $\mathcal{D}$ partition P.*
2. *If an optimal center $c_i^* \in C_i^*$ is contained in a cluster $D_\ell(c) \in \mathcal{D}$, then $\ell \leq i$.*

Next, for every non-empty subset $\mathcal{D}' \subseteq \mathcal{D}$, we call MatroidIndependentSet($\mathcal{D}', \mathcal{D}$). In this algorithm, we define a matroid $\mathcal{M}(\mathcal{D}', \mathcal{D})$ – see Algorithm 3 for the definition. It is easy to see that for any $\mathcal{D}' \subseteq \mathcal{D}$, that $\mathcal{M}(\mathcal{D}', \mathcal{D})$ is a (partition) matroid on P. We then find a common maximum independent set C in both matroids $\mathcal{M}$ and $\mathcal{M}(\mathcal{D}', \mathcal{D})$. Then, if $c \in C$ is contained in a cluster of level i in $\mathcal{D}$, then we place a ball of radius $9r_i$ around it. Next, we prove that, for at least one subset $\mathcal{D}' \subseteq \mathcal{D}$, the algorithm MatroidIndependentSet($\mathcal{D}', \mathcal{D}$) finds a set of balls that covers the entire point set P.

Algorithm 3 MatroidIndependentSet($\mathcal{D}', \mathcal{D}$).

1: Define a new matroid $\mathcal{M}(\mathcal{D}', \mathcal{D}) = (P, \mathcal{I}(\mathcal{D}'))$, where a set $C \subseteq P$ is independent in $\mathcal{I}(\mathcal{D}')$ iff it contains at most one point from each cluster in $\mathcal{D}'$, and no points from cluster in $\mathcal{D} \setminus \mathcal{D}'$
2: The weight of an independent set is equal to its size
3: Solve the maximum-weight matroid intersection problem for matroids $\mathcal{M}$ and $\mathcal{M}(\mathcal{D}', \mathcal{D})$ to find an independent set $C \subseteq P$
4: For every $c \in C$, place a ball of radius $9r_i$, if c is covered by a level i cluster in $\mathcal{D}$
5: **return** the resulting set of balls placed around each center in C

To this end, let $\mathcal{D}^* \subseteq \mathcal{D}$ denote the subset of clusters that contain at least one optimal center. This implies that clusters in $\mathcal{D} \setminus \mathcal{D}^*$ contain no optimal center. In the following claim, we focus on the call MatroidIndependentSet($\mathcal{D}^*, \mathcal{D}$), and show that the set of balls found in this call covers P.

▶ **Lemma 4.** *The set of balls computed by* MatroidIndependentSet($\mathcal{D}^*, \mathcal{D}$) *covers the entire set of points. Furthermore, the cost of this set of balls is upper bounded by* 9 *times the cost of* $\mathcal{B}$*.*

Proof. From every cluster in $\mathcal{D}^*$, pick an arbitrary optimal center, and let the resulting set be $\hat{C}^*$. Since $\hat{C}^* \subseteq C^*$, it is an independent set in $\mathcal{M}$. Furthermore, it contains exactly one point from each cluster in $\mathcal{D}^*$, and no point from any cluster in $\mathcal{D} \setminus \mathcal{D}^*$. Therefore, $\hat{C}^*$ is independent in the matroid $\mathcal{M}(\mathcal{D}^*, \mathcal{D})$.

Let C denote the maximum weight independent subset computed in MatroidIndependentSet($\mathcal{D}^*, \mathcal{D}$). Thus $|C| \geq |\hat{C}^*|$. As C is independent in $\mathcal{M}(\mathcal{D}^*, \mathcal{D})$, this implies that C (like $\hat{C}^*$) contains exactly one point from each cluster in $\mathcal{D}^*$.

We prove that the set of balls, centered at C, computed at the end of MatroidIndependentSet($\mathcal{D}^*, \mathcal{D}$), covers all points of P. Fix a point $p \in P$, and suppose it is covered by an optimal ball $B(c_j^*, r_j)$. From Observation 3, there exists a cluster $D_\ell(c) \in \mathcal{D}^*$ of level $\ell \leq j$ such that $c_j^* \in D_\ell(c)$. Therefore, $d(p, c) \leq r_j + 4r_\ell \leq 5r_\ell$. From the previous paragraph, there exists a center $c' \in C \cap D_\ell(c)$. Note that $d(c, c') \leq 4r_\ell$. This implies that, $d(p, c') \leq d(p, c) + d(c, c') \leq 5r_\ell + 4r_\ell = 9r_\ell$. Thus, p is covered by the ball of radius $9r_\ell$ centered at c'.

Finally, note that for every cluster in $\mathcal{D}^*$ of level i, we place at most one ball of radius $9r_i$. Therefore, the cost of the balls thus computed can be bounded by 9 times the cost of balls in $\mathcal{B}$. ◀

▶ **Theorem 5.** *For any fixed* $\epsilon \geq 0$*, there exists a* $(9 + O(\epsilon))$*-approximation algorithm to the Matroid Sum-of-radii problem that runs in* $b^{O(b)} \cdot n^{O(1)}$ *time, where* b *denotes the size of a base in the given matroid.*

Proof. We focus on a particular value of k', and show that the algorithm runs in $k'^{O(k')} \cdot n^{O(1)}$ time. Then, the running time guarantee follows, since $\sum_{k'=1}^{b} k'^{O(k')} = b^{O(b)}$, as previously discussed.

There are $k'^{O(\log k')}$ choices for guessing the "radius profile", and one of these choices corresponds to that of the modified optimal solution. Now fix this choice of the radius profile. At any level $1 \le i \le t$, there are at most $k'^{O(k_i)}$ recursive calls to the algorithm at level $i+1$. Therefore, the number of recursive calls at level $t+1$ can be bounded by $k'^{\sum_{i=1}^{t} O(k_i)} = k'^{O(k')}$. At level $t+1$, we call MatroidIndependentSet($\mathcal{D}', \mathcal{D}$) for every non-empty $\mathcal{D}' \subseteq \mathcal{D}$. Thus, there are at most $2^{|\mathcal{D}|} \le 2^{k'}$ calls to MatroidIndependentSet, and each matroid intersection problem can be solved in polynomial time, given access to the oracle for $\mathcal{M}$. Therefore, the algorithm terminates in $k'^{O(k')} \cdot n^{O(1)}$ time.

Now, consider the iteration when $k' = k$, and when we correctly guess the radius profile corresponding to the optimal solution. From Lemma 4, one of the calls to MatroidIndependentSet computes a solution that covers all the points, and the cost of this solution can be upper bound by 9 times the cost of the radius profile. Therefore, the cost of this solution can be upper bounded by $9 + O(\epsilon)$ times the cost of the *original* optimal solution. ◀

3 Uniform Capacitated Sum of Radii

Problem Definition

In this problem, we are given a finite metric space (P, d). We are also given a positive integer U, which denotes the *capacity*. We want to place a set of k balls $\mathcal{B}$, and assign each point of P to a ball containing it, such that no ball is assigned more than U points. Furthermore, we want to minimize the sum of radii of the balls in $\mathcal{B}$.

More formally, a feasible solution to the problem consists of a set of centers $C \subseteq P$ of size at most k, and a radius assignment $r : C \to \mathbb{R}^+$. Let $\mathcal{B}$ be the set of resulting balls. Note that the centers of balls in $\mathcal{B}$ are distinct. The solution also consists of an assignment $\mu : P \to \mathcal{B}$, such that p is contained in the ball $\mu(p)$, and $|\mu^{-1}(B)| \le U$ for every $B \in \mathcal{B}$. Finally, the objective is find such a feasible solution that minimizes the sum of radii: $\sum_{B(c,r) \in \mathcal{B}} r$.

Notation

Let $P' \subseteq P$ be a subset of points, and let $\mathcal{B}'$ be some non-empty set of balls. Then, an assignment $\mu : P' \to \mathcal{B}'$ is said to be a *valid* assignment, if it satisfies the following two properties: (i) for every $p \in P', p \in \mu(p)$, and (ii) $|\mu^{-1}(B)| \le U$ for any ball $B \in \mathcal{B}'$. In the following discussion, we will allow $\mathcal{B}'$ to contain concentric balls and even be a multi-set. But for simplicity of exposition, we will refer to a multi-set (resp. a multi-subset thereof) as simply a set (resp. a subset). Note that the definition of a valid assignment is consistent even if $\mathcal{B}$ is such a set of balls, by treating each copy of a ball in $\mathcal{B}$ as a distinct object.

Fix an optimal solution and the corresponding optimal assignment. We preprocess this solution in order to discretize the set of radii, exactly as done in the previous section. After this discretization, we assume that the solution uses exactly k_i balls of radius r_i, where $i \le i \le t$, and $t = O(\log k)$. Henceforth we will refer to the optimal solution modified in this manner. As in the previous section, let $\mathcal{O}$ denote the set of optimal balls, and $C^* \subseteq P$ be the set of optimal centers. The subsets $\mathcal{O}_i, \mathcal{O}_{\le i}$ etc. of the optimal balls $\mathcal{O}$, and the subsets $C^*_i, C^*_{\le i}$ etc. of the optimal centers are also defined as in the previous section. Furthermore, let $\mu_* : P \to \mathcal{O}$ be the optimal assignment. Note that μ_* is a valid assignment by definition.

Algorithm

Our algorithm invokes CapacitatedSoR($\emptyset$, 1). Now we describe CapacitatedSoR($\mathcal{B}, i$) (Algorithm 4), which is recursive and proceeds in multiple levels. At a particular level $1 \le i \le t$, we determine the set of balls of level i in the solution. At the start of the algorithm at level i, we are given a (multi-)set $\mathcal{B}$ of balls chosen earlier. $\mathcal{B}$ consists of balls of level 1 through $i-1$. Before we discuss the algorithm in iteration i, let us define some more notation.

Suppose a ball centered at c was added to $\mathcal{B}$ at level $j < i$ – its radius was $6r_j$ when it was added to $\mathcal{B}$. At every subsequent iteration $j+1 \le \ell < i$, we expand its radius by an additive $2r_\ell$ factor. Thus, at the beginning of iteration i, the radius of this ball is $6r_j + \sum_{\ell=j+1}^{i-1} 2r_\ell$. Now, in iteration i, we will consider two versions of any ball in $\mathcal{B}$ – expanded and unexpanded. Consider a ball in $\mathcal{B}$, with center c, added during iteration $j < i$. At the beginning of iteration i, this ball has radius $6r_j + \sum_{\ell=j+1}^{i-1} 2r_\ell$; we refer to this as the unexpanded version. On the other hand, the expanded version has radius equal to $6r_j + \sum_{\ell=j+1}^{i} 2r_\ell$, which we denote by $E_j^i(c)$. Note that the expanded version $E_j^i(c)$ is larger than its corresponding version $B_j^i(c)$ by an additive $+2r_i$ factor. Therefore, if $B^* = B(c_i^*, r_i)$ has a non-empty intersection with $B_j^i(c)$, then $B^* \subseteq E_j^i(c)$.

Let $\mathcal{B}' \subseteq \mathcal{B}$ be any subset of balls chosen so far, and let $\overline{\mathcal{B}'} = \mathcal{B} \setminus \mathcal{B}'$. Let $E(\mathcal{B}')$ denote the set of expanded versions of balls in $\mathcal{B}'$. Finally, we define

$$\mathcal{I}(\mathcal{B}') := \begin{cases} \left(\bigcap_{E \in E(\mathcal{B}')} E\right) \setminus \left(\bigcup_{B \in \overline{\mathcal{B}'}} B\right) & \text{if } \mathcal{B}' \neq \emptyset \\ P \setminus \left(\bigcup_{B \in \mathcal{B}} B\right) & \text{if } \mathcal{B}' = \emptyset \end{cases}$$

That is, if $\mathcal{B}'$ is non-empty, then $\mathcal{I}(\mathcal{B}')$ is exactly the set of points that belong to the common intersection of the expanded versions of balls in $\mathcal{B}'$, but not in any of the unexpanded versions of balls in $\overline{\mathcal{B}'}$. If $\mathcal{B}'$ is empty, then $\mathcal{I}(\mathcal{B}')$ is the set of points that does not belong to any unexpanded ball in $\mathcal{B}$. Note that if $\mathcal{B}' \subseteq \mathcal{B}$ is exactly the subset of balls that have non-empty intersection with an optimal ball $B(c_i^*, r_i)$, then $B(c_i^*, r_i) \subseteq \mathcal{I}(\mathcal{B}')$.

Let us return to the discussion of CapacitatedSoR($\mathcal{B}, i$). For each subset $\mathcal{B}' \subseteq \mathcal{B}$, we call Greedy($\mathcal{B}', \mathcal{B}, r_i$) (Algorithm 5). This Algorithm computes a set $P_i(\mathcal{B}')$ of at most $4k$ centers chosen in a certain "greedy" manner. This is the set of potential centers from the region $\mathcal{I}(\mathcal{B}')$ for placing balls of level i. The algorithm ensures that the distance between any two centers in $P_i(\mathcal{B})$ is greater than $4r_i$. Furthermore, if $P_i(\mathcal{B}') < 4k$, then each point in $\mathcal{I}(\mathcal{B}')$ is within distance $4r_i$ of some point in $P_i(\mathcal{B}')$. We repeat this process for every subset $\mathcal{B}'$ of $\mathcal{B}$ (including $\emptyset$).

We will first argue in Lemma 6 that, in some recursive call to the algorithm at level $i = t+1$, the set of balls $\mathcal{B}$ computed captures the optimal solution in an appropriate way. Having shown that this happens, the invocation of PostProcess($\mathcal{B}$) (Algorithm 7) will appropriately modify the set of balls $\mathcal{B}$ and return a feasible solution. We will discuss the algorithm PostProcess and its analysis later.

▶ **Lemma 6.** *Fix a level $1 \le i \le t+1$. In one of the recursive calls to* CapacitatedSoR($\mathcal{B}, i$)*, there exists an assignment that maps each point $p \in \mu_*^{-1}(\mathcal{O}_{<i})$ to a ball in $\mathcal{B}$ containing p, such that the number of points assigned to each ball does not exceed U.*

Proof. We prove this lemma inductively.

Algorithm 4 CAPACITATEDSOR($\mathcal{B}, i$).

if $i = t + 1$ **then**
  if POSTPROCESS($\mathcal{B}$) $\neq$ **fail then**
    return $\mathcal{B}(R)$ returned by POSTPROCESS($\mathcal{B}$) and **halt**
else
  For every $\mathcal{B}' \subseteq \mathcal{B}$, let $P_i(\mathcal{B}') \leftarrow$ GREEDY($\mathcal{B}', \mathcal{B}, r_i$)
  Let $P_i \leftarrow \bigcup_{\mathcal{B}' \subseteq \mathcal{B}} P_i(\mathcal{B}')$
  for every multi-subset $C_i \subseteq P_i$ of size at most k_i **do**
    Expand every ball in $\mathcal{B}$ by an additive $2r_i$ factor
    CAPACITATEDSOR($\mathcal{B} \cup \mathcal{B}(C_i), i + 1$) ▷ $\mathcal{B}(C_i) \coloneqq \{B(c, 6r_i) : c \in C_i\}$

Algorithm 5 GREEDY($\mathcal{B}', \mathcal{B}, r$).

Let $T \leftarrow \mathcal{I}(\mathcal{B}')$; start with all points of T as unmarked
$P(\mathcal{B}') \leftarrow \emptyset$
while $|P(\mathcal{B}')| < 4k$ and there is an unmarked point in T **do**
  $p \in T$ be an unmarked point with maximum number of unmarked points in $B(p, r) \cap T$
  Add p to $P(\mathcal{B}')$; mark all points in $B(p, 4r) \cap T$
return $P(\mathcal{B}')$

Base case

This corresponds to the start of the calls CAPACITATEDSOR($\cdot, i$), where $i = 2$. To this end, consider the invocation of the algorithm at the earlier level, i.e., CAPACITATEDSOR($\emptyset, 1$). Note that since $\mathcal{B} = \emptyset$, $\mathcal{B}' = \emptyset$, and there is only one call GREEDY($\emptyset, \emptyset, r_1$). Note that the optimal solution uses $k_1 + k_2 + \ldots + k_t \leq k$ balls of radius at most r_1 in order to cover the entire set of points $P = \mathcal{I}(\mathcal{B}')$. Recall that the set of centers $P_1(\emptyset) = P_1$ returned by the Greedy algorithm has the property that any two centers in $P_1(\emptyset)$ are at least $4r_1$ away from each other. Therefore, $P_1(\emptyset)$ contains at most one point from each optimal ball, and thus $|P_1(\emptyset)| \leq k$. It follows that for any optimal center $c_1^* \in C_1^*$, there is a center $c \in P_1$, such that $d(c_1^*, c) \leq 4r_1$. That is, for every optimal ball $B^* = B(c_1^*, r_1) \in \mathcal{O}_1$, there exists $c \in P_1$, such that $B^* \subseteq B(c, 5r_1)$. We let $\varphi(B^*) \coloneqq c$ (select a nearest c form c_1^* there are multiple such $c \in P_1$). Let $C_1 \subseteq P_1$ be the multi-set that is the image of the mapping $\varphi : \mathcal{O}_1 \to P_1$, where the multiplicity of each $c \in C_1$ is equal to $|\varphi^{-1}(c)|$. Therefore, for each $B^* \in \mathcal{O}_1$ the points in $\mu_*^{-1}(B^*)$ can be reassigned to a unique ball in $\mathcal{B}(C_1)$ centered at $\varphi(B^*) \in C_1$. This completes the proof for the base case.

Note that we were able to "guess" the locations of the optimal centers approximately in the base case. However, we cannot accomplish this in the subsequent levels, because some optimal balls may be contained in larger optimal balls. This is what complicates the algorithm and its analysis. Nevertheless, we will argue that we can find an appropriate set of substitute centers whenever necessary that will facilitate the reassignment process.

Inductive hypothesis

Suppose during some invocation of the algorithm CAPACITATEDSOR($\mathcal{B}, i$) at level i, we have a set of balls $\mathcal{B}$ of levels $1 \leq j < i$, such that the set of points $\mu_*^{-1}(\mathcal{O}_{<i})$ can be assigned to the balls in $\mathcal{B}$. Let us also suppose that we have a mapping $\varphi : \mathcal{O}_{<i} \to C_{<i}$, where $C_{<i}$ denotes the set of centers of balls in $\mathcal{B}$.

Inductive Step

We first sketch the high level idea. Here, we will extend φ to include $\mathcal{O}_i$, i.e., we will map every optimal ball in $\mathcal{O}_i$ to a center in P_i; where P_i is the set of centers computed in lines 5 and 6 of Algorithm 4. Now, let $C_i \subseteq P_i$ be the image of $\varphi(\mathcal{O}_i)$, where the multiplicty of each $c \in P_i$ is set to be $|\varphi^{-1}(c)|$. Note that $|C_i| = |\mathcal{O}_i| = k_i$. Having found such a mapping, we will consider an optimal ball $B_i^* = B(c_i^*, r_i) \in \mathcal{O}_i$, and reassign points in $\mu_*^{-1}(B_i^*)$ to the balls in $\mathcal{B} \cup \mathcal{B}(C_i)$. We will use the ball $\varphi(B_i^*)$ to show that this reassignment can be done without violating the capacities. Doing this for every optimal ball in $\mathcal{O}_i$, we will show that all points in $\mu_*^{-1}(\mathcal{O}_{\le i})$ are assigned to balls in $\mathcal{B} \cup \mathcal{B}(C_i)$. Now we discuss the details of this inductive argument.

For any $B^* = B(c_i^*, r_i) \in \mathcal{O}_i$, if there is $c \in P_i$ such that $B^* \subseteq B(c, 6r_i)$, then we set $\varphi(B_i^*) = c$ (choosing a nearest such c from c^*, if there are multiple such $c^* \in P_i$). Let $\mathcal{O}_i^1 \subseteq \mathcal{O}_i$ be the subset that is mapped in such way, and let $C_i^1 \subseteq P_i$ be its image (with multiplicity $\varphi^{-1}(c)$ for every $c \in C_i^1$). Note that all the points assigned a ball $B_i^* \in \mathcal{O}_i^1$, are also contained in the ball $B(c, 6r_i)$, where $c = \varphi(B_i^*)$. Therefore, we can reassign points in $\mu_*^{-1}(B_i^*)$ to the ball $B(c, 6r_i)$.

Now, let $\mathcal{O}_i^2 \coloneqq \mathcal{O}_i \setminus \mathcal{O}_i^1$ be the set of optimal balls not mapped so far. We will map each ball in $\mathcal{O}_i^2$ to a unique center in $P_i \setminus C_i^1$, and use this mapping to compute the required reassignment. We describe the assignment in the following mapping procedure – note that this is used only in the analysis.

Algorithm 6 Mapping procedure.

1: Suppose all balls in $\mathcal{O}_i^2$ are unmapped at the beginning; let $C_i^2 \leftarrow \emptyset$
2: **for** each subset $\mathcal{B}' \subseteq \mathcal{B}$ in an arbitrary order **do**
3: Let $\mathcal{O}_i^2(\mathcal{B}') \subseteq \mathcal{O}_i^2$ be the subset of *unmapped* balls contained in $\mathcal{I}(\mathcal{B}')$
▷ *Unmapped balls in $\mathcal{O}_i^2$ are the balls that have not yet been mapped using φ in an earlier iteration.*
4: Let $F_i(\mathcal{B}') \subseteq P_i(\mathcal{B}')$ include every point that:
(i) belongs to C_i^1, or (ii) is chosen as a center of a ball in $\mathcal{B}$, or
(iii) is within $2r_i$ from some center in C_i^2, or
(iv) is within $r_i + r_j$ from some center in C_j^*, where $j \ge i$, or
(v) belongs to $C_{<i}^*$.
5: Extend φ to $\mathcal{O}_i^2(\mathcal{B}')$ by arbitrarily mapping each ball in $\mathcal{O}_i^2(\mathcal{B}')$ to a unique center in $P_i(\mathcal{B}') \setminus F_i(\mathcal{B}')$.
6: Let $C_i^2(\mathcal{B}')$ be the image of $\mathcal{O}_i^2(\mathcal{B}')$, under the above mapping φ.
7: Mark all balls in $\mathcal{O}_i^2(\mathcal{B}')$ as mapped, and add $C_i^2(\mathcal{B}')$ to C_i^2.

▷ Claim 7. For any $\mathcal{B}' \subseteq \mathcal{B}$, if $\mathcal{O}_i^2(\mathcal{B}') \neq \emptyset$, then $|\mathcal{O}_i^2(\mathcal{B}')| \le |P_i(\mathcal{B}') \setminus F_i(\mathcal{B}')|$. That is, there are enough centers available in $P_i(\mathcal{B}') \setminus F_i(\mathcal{B}')$ to be mapped in Line 5.

Proof. Since $\mathcal{O}_i^2(\mathcal{B}') \neq \emptyset$, let $B(c_i^*, r_i) \in \mathcal{O}_i^2(\mathcal{B}')$. Note that $c_i^* \in \mathcal{I}(\mathcal{B}')$, and $B(c_i^*, r_i) \subseteq \mathcal{I}(\mathcal{B}')$.

Consider the call Greedy$(\mathcal{B}', \mathcal{B}, r_i)$. We first claim that the while loop ends with $|P_i(\mathcal{B}')| = 4k$. Suppose for the contradiction that the while loop ends because all points in $P_i(\mathcal{B}')$ are marked. Let c be the point added to $P_i(\mathcal{B}')$ when c_i^* is marked. Then, $d(c_i^*, c) \le 4r_i$. Thus, $B(c_i^*, r_i) \subseteq B(c, 6r_i)$, which implies that $B(c_i^*, r_i) \in \mathcal{O}_i^1$. This is a contradiction, since $B(c_i^*, r_i) \in \mathcal{O}_i^2$.

Now we claim that $|F_i(\mathcal{B}')| \le 3k$, by considering each of the five conditions ($F_i(\cdot)$ stands for centers *forbidden* due to one of the five conditions). Conditions (i) and (ii) include at most $\sum_{j=1}^{i-1} k_j$, and at most k_i points respectively. Therefore, k is an upper bound for points satisfying conditions (i) and (ii).

We now claim that k_i is also an upper bound for points satisfying condition (iii). To this end, we claim that for $c \in C_i^2$, there is at most one $c \in P_i(\mathcal{B}')$ such that $d(c, c') \leq 2r_i$. Suppose that there are two distinct such points $c_1, c_2 \in P_i(\mathcal{B}')$. Then, $d(c_1, c_2) \leq d(c, c_1) + d(c, c_2) \leq 4r_i$. This is a contradiction, since the distance between any two points in $P_i(\mathcal{B}')$ is greater than $4r_i$. Finally, since $|C_i^2| \leq k$, k is also an upper bound on the centers excluded due to condition (iii).

A similar proof also shows that for any fixed $c_j^* \in C_j^*$ with $j \geq i$, there is at most one $c \in P_i(\mathcal{B}')$ with $d(c_j^*, c) \leq r_i + r_j$. Therefore, $\sum_{j=i}^{t} k_j$ is an upper bound for points satisfying condition (iv). $\sum_{j=1}^{i-1} k_j$ is an upper bound on condition (v). Therefore, k is an upper bound on conditions (iv) and (v) together.

Putting everything together, $|F_i(\mathcal{B}')| \leq 3k$, which implies that $|P_i(\mathcal{B}') \setminus F_i(\mathcal{B}')| \geq k$. Therefore, each ball in $\mathcal{O}_i^2(\mathcal{B}')$ can be mapped to a unique point in $|P_i(\mathcal{B}')|$. ◁

The next claim is used later to argue that the reassignment can be done using the mapping φ constructed in this manner.

▷ Claim 8. Fix $\mathcal{B}' \subseteq \mathcal{B}$ and a ball $B^* = B(c_i^*, r_i) \in \mathcal{O}_i^2(\mathcal{B}')$. If $\varphi(B^*) = c$, then $|B(c, r_i) \cap \mathcal{I}(\mathcal{B}')| \geq |B^*| \geq |\mu_*^{-1}(B^*)|$.

Proof. We first claim that no point in B^* is marked in Greedy($\mathcal{B}', \mathcal{B}, r_i$). Otherwise, let $c' \in \mathcal{I}(B^*)$ be the point added to $P_i(\mathcal{B}')$ when a point $p \in B^*$ was marked. Then, $d(c', c_i^*) \leq d(c', p) + d(p, c_i^*) \leq 5r_i$, which implies that $B^* \subseteq B(c', 6r_i)$, which implies that $B(c_i^*, r_i) \in \mathcal{O}_i^1$. This is a contradiction, since $B(c_i^*, r_i) \in \mathcal{O}_i^2$. Therefore no point in $B^* \cap \mathcal{I}(\mathcal{B}') = B^*$ is marked until the end of the while loop.

Now, consider the beginning of the iteration when c was added to $P_i(\mathcal{B}')$. At this point, c_i^* is also a candidate. Since c is chosen over c_i^*, it implies that $|B(c, r_i) \cap \mathcal{I}(\mathcal{B}')| \geq |B^*| \geq |\mu_*^{-1}(B^*)|$, where the last inequality holds by definition. ◁

We use this claim to show that we can reassign points from $\mu_*^{-1}(\mathcal{O}_i^2)$ to balls in $\mathcal{B} \cup \mathcal{B}(C_i)$, where $C_i = C_i^1 \cup C_i^2$. Recall that we have already reassigned points $\mu_*^{-1}(\mathcal{O}_i^1)$ to C_i^1.

Now let us consider the optimal balls in $\mathcal{O}_i^2$ in the same order in which they were mapped in Algorithm 6. Consider an optimal ball $B^* = B(c_i^*, r_i)$, and suppose it was mapped in the iteration corresponding to $\mathcal{B}' \subseteq \mathcal{B}$. That is, $B^* \in \mathcal{O}_i^2(\mathcal{B}')$. Let $c = \varphi(B^*)$. Because of condition (v), there is no optimal center $c_j^* \in C_j^*$ with $j \geq i$, such that $B(c_j^*, r_j) \cap B(c, r_i) \neq \emptyset$. Therefore, all points in $B = B(c, r_i)$ are assigned to balls in $\mathcal{O}_{<i}$ in the optimal assignment μ_*. Furthermore, because of condition (iv), there is no other center $c' \in C_i^2$ within distance $2r_i$ from c, which implies that points in B have not been currently assigned to a ball in $\mathcal{B}(C_i)$. Therefore, by the inductive hypothesis, these points are assigned to balls in $\mathcal{B}$.

Now we reassign m points from the set $B \cap \mathcal{I}(\mathcal{B}')$ to the ball $B(c, 6r_i)$, where $m = \min\{U, |B \cap \mathcal{I}(\mathcal{B}')|\}$. These points are originally assigned to balls in $\mathcal{B}$. As they are contained in $\mathcal{I}(\mathcal{B}')$, no such point belongs to a ball in $\mathcal{B} \setminus \mathcal{B}'$, by the definition of $\mathcal{I}(\mathcal{B}')$. Thus, these points are assigned to balls in $\mathcal{B}'$. Their reassignment to $B(c, 6r_i)$ collectively frees up m units of capacity from balls in $\mathcal{B}'$. Note that B^* is also completely contained in $\mathcal{I}(\mathcal{B}')$, and $m \geq |\mu_*^{-1}(B^*)|$ by Claim 8. Therefore, we can use the freed capacity of balls in $\mathcal{B}'$ to assign points in $\mu_*^{-1}(B^*)$.

We perform this reassignment process for each ball in $\mathcal{O}_i^2$. Therefore, at the end, every point in $\mu_*^{-1}(\mathcal{O}_{\leq i})$ is assigned to a ball in $\mathcal{B} \cup \mathcal{B}(C_i)$. This finishes the proof of Lemma 6. ◀

Using Lemma 6 at level $t+1$, we know that there exists a recursive call to $\textsc{CapacitatedSoR}(\mathcal{B}, t+1)$, such that the set of points in $\mu_*^{-1}(\mathcal{O}) = P$ can be assigned to the balls in $\mathcal{B}$ without violating capacities. Fix such a recursive call. With this, let us define $C^1 = \bigcup_{i=1}^t C_i^1$, and $C^2 = \bigcup_{i=1}^t C_i^2$, and let $C = C^1 \cup C^2$. Note that there may be several concentric balls in $\mathcal{B}$. We want to move the concentric balls to "nearby" unique centers in order to obtain a feasible solution. The following observations, which follow from the description of the mapping procedure (see Algorithm 6), will aid us in doing this.

▶ **Observation 9.**

1. *For any $c \in C_i^1$, define $R^*(c)$ to be the set of optimal centers of the balls in $\varphi^{-1}(c)$.*
   (A) The sets $\{R^*(c)\}_{c \in C^1}$ are pairwise disjoint, i.e., for distinct $c_1, c_2 \in C^1$, we have that $R^*(c_1) \cap R^*(c_2) = \emptyset$.
   (B) $R^*(c) \subseteq B(c, 5r_i)$ for any $c \in C_i^1$.
2. *For any $c \in C^2$, define $R^*(c) \coloneqq \{c\}$*
   (A) $|\varphi^{-1}(c)| = 1$ for all $c \in C^2$
   (B) $C^2 \cap C^* = \emptyset$, and
   (C) $C^2 \cap C^1 = \emptyset$.
3. *Items 1 and 2 imply that the sets $\{R^*(c)\}_{c \in C}$ are pairwise disjoint.*

Proof. For item 1.A, note that $\varphi : \mathcal{O} \to C$ is a many-to-one function, and that every ball in $\mathcal{O}$ has a distinct center. Item 1.B follows from the definition of φ.

Claims in item 2 follow from the definition of set of *forbidden* centers $F_i(\mathcal{B}')$ in the mapping procedure (see line 4). ◀

Now we are ready to show that, when $\textsc{PostProcess}(\mathcal{B})$ (Algorithm 7) is called from $\textsc{CapacitatedSoR}(\mathcal{B}, t+1)$, where $\mathcal{B}$ is the set of balls guaranteed by Lemma 6, it successfully returns a feasible solution.

Algorithm 7 $\textsc{PostProcess}(\mathcal{B})$.

1: For every $1 \le i \le t$, and every $c \in C_i$, find a set $R(c) \subseteq B(c, 5r_i)$ where $|R(c)|$ equals the multiplicity of c in C_i and the sets $R(c)$ are pairwise disjoint for all $c \in C$
   This can be solved using a max-flow problem
2: **if** such a collection of sets $R(c)$ does not exist: **return fail**
3: Let $R = \bigcup_{c \in C} R(c)$, and let $\mathcal{B}(R) \coloneqq \{B(c, \alpha \cdot r_i) : c \in R\}$
   ▷ *α is defined below in the proof of Lemma 10*
4: Check whether there exists a feasible assignment from P to the balls in $\mathcal{B}(R)$
   This can be solved using a max-flow problem
5: **if** a feasible assignment exists: **return** $\mathcal{B}(R)$; **else**: **return fail**

▶ **Lemma 10.** $\textsc{PostProcess}(\mathcal{B})$ *succeeds in finding a set of balls $\mathcal{B}(R)$, and there is a feasible assignment $\mu' : P \to \mathcal{B}(\mathcal{R})$.*

Proof. From Lemma 6, there exists a feasible assignment $\mu : P \to \mathcal{B}$, however there may be concentric balls in the set $\mathcal{B}$. However, Observation 9 implies that the sets $R^*(c)$ are pairwise disjoint for $c \in C$, and that $R^*(c) \subseteq B(c, 5r_i)$ for a center $c \in C_i$. Therefore, in line 1 of Algorithm $\textsc{PostProcess}(\mathcal{B})$, we can successfully find the sets $R(c)$ as claimed. Note that $R(c) \subseteq B(c, 5r_i)$.

From Lemma 6, for any $B = B(c, 6r_i) \in \mathcal{B}$, $\mu^{-1}(B) \subseteq E_i^t(c)$. Note that the radius of the expanded version of the ball $E_i^t(c)$ is equal to $6r_i + \sum_{\ell=i+1}^{t} 2r_\ell \leq \alpha' \cdot r_i$, for some α'. Therefore, for any point $p \in \mu^{-1}(B)$, and any $c' \in R(c)$, we have that $d(p, c') \leq 5r_i + \alpha' r_i = (\alpha' + 5)r_i = \alpha r_i$ [1]. This implies that, in line 4 of the Algorithm 7, we can find such a feasible assignment μ'. ◀

▶ **Lemma 11.** CapacitatedSoR$(\emptyset, 1)$ *runs in* $2^{O(k^2)} \cdot n^{O(1)}$ *time.*

Proof. Fix a level $1 \leq i \leq t$, and consider CapacitatedSoR$(\mathcal{B}, i)$. At the beginning of the algorithm, $|\mathcal{B}| \leq k$, therefore the number of subsets can be upper bound by 2^k, which implies that $|P_i| \leq 4k \cdot 2^k$. Now, let $k_i' \leq k_i$ denote the size of the set C_i without multiplicities. Therefore, there are $\sum_{k_i'=1}^{k_i} \binom{4k \cdot 2^k}{k_i'} = \left(4k \cdot 2^k\right)^{O(k_i)} = 2^{O(k \cdot k_i)}$ number of choices for selecting the set C_i (without multiplicities). For a fixed choice of C_i, there are at most $(k_i)^{k_i'} = k^{O(k_i)}$ choices for placing one or more copies at each location in C_i. We make a recursive call for each such choice of the multi-set C_i. The overall number of recursive calls to level $i+1$ can be upper bounded by $k^{O(k_i)} \cdot 2^{O(k \cdot k_i)} = 2^{O(k \cdot k_i)}$.

Let $T(i)$ denote the running time of the algorithm at level i. Then, we have the following recurrence relation: $T(i) = 2^{O(k \cdot k_i)} \cdot T(i+1) + 2^{O(k \cdot k_i)} \cdot n^{O(1)}$.
Furthermore, $T(t+1) = n^{O(1)}$, since PostProcess runs in time polynomial in n. This recurrence solves to $T(1) = 2^{O(k^2)} \cdot n^{O(1)}$, where we use the fact that $\sum_{i=1}^{t} k_i = k$. ◀

▶ **Theorem 12.** *There exists a 28-approximation for the Capacitated Sum of Radii problem that runs in* $2^{O(k^2)} \cdot n^{O(1)}$ *time.s*

Proof. There are $O(n^2)$ choices for guessing the maximum radius, and $k^{O(\log k)}$ choices for guessing the radius profile of the optimal solution. Note that we lose a factor of $(1+\epsilon)^2$ in the latter step. Now, fix the correct value of maximum radius and the radius profile that corresponds to the modified optimal solution. By Lemma 11, the algorithm CapacitatedSoR$(\emptyset, 1)$ runs in $2^{O(k^2)} \cdot n^{O(1)}$ time for any fixed choice of the radius profile.

Furthermore, By Lemma 10, there exists a recurse call at level $t+1$, that returns a feasible solution. Finally, for any $1 \leq i \leq t$ we use k_i balls of radius αr_i in this solution, whereas the optimal solution uses k_i balls of radius r_i. Therefore, the approximation guarantee is at most $(1+\epsilon)^2 \cdot \alpha$, where α is as in Lemma 10. Choosing $\epsilon \approx 0.267$, the above quantity can be upper bounded by 28. ◀

4 Conclusion

We obtain constant approximations for the uniform capacitated sum of radii problem in FPT time. It is unclear whether a similar result can be obtained in polynomial time. Finally, obtaining a constant approximation for the matroid version of the problem in polynomial time remains open.

[1] It can be shown that $\alpha = \frac{2+11\cdot\epsilon(1+\epsilon)}{\epsilon(1+\epsilon)}$.

References

1 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed M. Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany.*, pages 1:1–1:14, 2019. doi:10.4230/LIPIcs.ESA.2019.1.

2 Hyung-Chan An, Mohit Singh, and Ola Svensson. Lp-based algorithms for capacitated facility location. *SIAM J. Comput.*, 46(1):272–306, 2017. doi:10.1137/151002320.

3 Sayan Bandyapadhyay, Santanu Bhowmick, Tanmay Inamdar, and Kasturi R. Varadarajan. Capacitated covering problems in geometric spaces. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, volume 99 of *LIPIcs*, pages 7:1–7:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.SoCG.2018.7.

4 Babak Behsaz and Mohammad R. Salavatipour. On minimum sum of radii and diameters clustering. *Algorithmica*, 73(1):143–165, 2015. doi:10.1007/s00453-014-9907-3.

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

6 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. doi:10.1007/978-3-319-33461-5_22.

7 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. doi:10.1006/jcss.2002.1882.

8 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.

9 Danny Z Chen, Jian Li, Hongyu Liang, and Haitao Wang. Matroid and knapsack center problems. *Algorithmica*, 75(1):27–52, 2016.

10 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. URL: http://dl.acm.org/citation.cfm?id=1070432.1070569.

11 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, July 9-12, 2019, Patras, Greece.*, pages 41:1–41:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.41.

12 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

13 H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with (1+epsilon)-capacity violation. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.73.

14 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.

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

16 MohammadTaghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Budgeted red-blue median and its generalizations. In *European Symposium on Algorithms*, pages 314–325. Springer, 2010.

17 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.

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

19 Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 646–659, 2018.

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

21 Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. *ACM Trans. Algorithms*, 13(2):22:1–22:18, 2017. doi:10.1145/2983633.