A Lottery Model for Center-type Problems With Outliers*

David G. Harris[1], Thomas Pensyl[2], Aravind Srinivasan[3], and Khoa Trinh[4]

1 **Department of Computer Science, University of Maryland, College Park, USA**
`davidgharris29@gmail.com`

2 **Department of Computer Science, University of Maryland, College Park, USA**
`tpensyl@cs.umd.edu`

3 **Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, USA**
`srin@cs.umd.edu`

4 **Department of Computer Science, University of Maryland, College Park, USA**
`khoa@cs.umd.edu`

Abstract

In this paper, we give tight approximation algorithms for the k-center and matroid center problems with outliers. Unfairness arises naturally in this setting: certain clients could always be considered as outliers. To address this issue, we introduce a lottery model in which each client j is allowed to submit a parameter $p_j \in [0, 1]$ and we look for a random solution that covers every client j with probability at least p_j. Our techniques include a randomized rounding procedure to round a point inside a matroid intersection polytope to a basis plus at most one extra item such that all marginal probabilities are preserved and such that a certain linear function of the variables does not decrease in the process with probability one.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics

Keywords and phrases approximation algorithms, randomized rounding, clustering problems

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

The classic k-center and Knapsack Center problems are known to be approximable to within factors of 2 and 3 respectively [5]. These results are best possible unless P=NP [6, 5]. In these problems, we are given a metric graph G and want to find a subset $\mathcal{S}$ of vertices of G subject to either a cardinality constraint or a knapsack constraint such that the maximum distance from any vertex to the nearest vertex in $\mathcal{S}$ is as small as possible. We shall refer to vertices in G as *clients*. Vertices in $\mathcal{S}$ are also called *centers*.

It is not difficult to see that a few *outliers* (i.e., very distant clients) may result in a very large optimal radius in the center-type problems. This issue was raised by Charikar et. al. [2], who proposed a *robust* model in which we are given a parameter t and only need to serve t out of given n clients (i.e. $n - t$ *outliers* may be ignored in the solution). Here we consider three robust center-type problems: the Robust k-Center (RkCenter) problem, the Robust

* Research supported in part by NSF Awards CNS-1010789 and CCF-1422569, and by a research award from Adobe, Inc.

Knapsack Center (RKnapCenter) problem, and the Robust Matroid Center (RMatCenter) problem.

Formally, an instance $\mathcal{I}$ of the RkCenter problem consists of a set V of vertices, a metric distance d on V, an integer k, and an integer t. Let $n = |V|$ denote the number of vertices (clients). The goal is to choose a set $\mathcal{S} \subseteq V$ of centers (facilities) such that (i) $|\mathcal{S}| \leq k$, (ii) there is a set of *covered* vertices (clients) $\mathcal{C} \subseteq V$ of size at least t, and (iii) the objective function

$$R := \max_{j \in \mathcal{C}} \min_{i \in \mathcal{S}} d(i, j)$$

is minimized.

In the RKnapCenter problem, we are given a budget $B > 0$ instead of k. In addition, each vertex $i \in V$ has a weight $w_i \in \mathbb{R}_+$. The cardinality constraint (i) is replaced by the knapsack constraint: $\sum_{i \in \mathcal{S}} w_i \leq B$. Similarly, in the RMatCenter problem, the constraint (i) is replaced by a matroid constraint: $\mathcal{S}$ must be an independent set of a given matroid $\mathcal{M}$. Here we assume that we have access to the rank oracle of $\mathcal{M}$.

In [2], the authors introduced a greedy algorithm for the RkCenter problem that achieves an approximation ratio of 3. Recently, Chakrabarty et. al. [1] give a 2-approximation algorithm for this problem. Since the k-center problem is a special case of the RkCenter problem, this ratio is best possible unless P=NP.

The RKnapCenter problem was first studied by Chen et. al. [3]. In [3], the authors show that one can achieve an approximation ratio of 3 if allowed to slightly violate the knapsack constraint by a factor of $(1 + \epsilon)$. It is still unknown whether there exists a true approximation algorithm for this problem. The current inapproximability bound is still 3 due to the hardness of the Knapsack Center problem.

The current best approximation guarantee for the RMatCenter problem is 7 by Chen et. al. [3]. This problem has a hardness result of $(3 - \epsilon)$ via a reduction from the k-supplier problem.

From a practical viewpoint, unfairness arises inevitably in the robust model: some clients will always be considered as outliers and hence not covered within the guaranteed radius. To address this issue, we introduce a *lottery model* for these problems. The idea is to randomly pick a solution from a *public list* such that each client $j \in V$ is guaranteed to be covered with probability at least p_j, where $p_j \in [0, 1]$ is the success rate requested by j. In practice, one possible way to determine these p_j's is based on the cost that the clients are willing to pay for their probability of being served. Also, observe that the special case when $p_j = 1$ for all $j \in V$ is equivalent to the standard model.

In this paper, we introduce new approximation algorithms for these problems under this model. (Note that this model has been used recently for the k-center and Knapsack Center problems (without outliers) in [4], which will appear soon on arXiv. All the techniques and problems in [4] are different.) We also propose improved approximation algorithms for the RkCenter problem and the RMatCenter problem.

1.1 The Lottery Model

In this subsection, we formally define our lottery model for the above-mentioned problems. First, the *Fair* Robust k-Center (FRkCenter) problem is formulated as follows. Besides the parameters V, d, k and t, each vertex $j \in V$ has a "target" probability $p_j \in [0, 1]$. We are interested in the minimum radius R for which there exists a distribution $\mathcal{D}$ on subsets of V such that a set $\mathcal{S}$ drawn from $\mathcal{D}$ satisfies the following constraints:

Coverage constraint: $|\mathcal{C}| \geq t$ with probability one, where $\mathcal{C}$ is the set of all clients in V that are within radius R from some center $\mathcal{S}$,

Fairness constraint: $\Pr[j \in \mathcal{C}] \geq p_j$ for all $j \in V$, where $\mathcal{C}$ is as in the coverage constraint,

Cardinality constraint: $|\mathcal{S}| \leq k$ with probability one.

Here we aim for a polynomial-time, randomized algorithm that can sample from $\mathcal{D}$. Note that the RkCenter is a special of this variant in which all p_j's are set to be zero.

The *Fair* Robust Knapsack Center (FRKnapCenter) problem and *Fair* Robust Matroid Center (FRMatCenter) problem are defined similarly except that we replace the cardinality constraint by a knapsack constraint and a matroid constraint, respectively. More formally, in the FRKnapCenter problem, we are given a budget $B \in \mathbb{R}^+$ and each vertex i has a weight $w_i \in \mathbb{R}^+$. We require the total weight of centers in $\mathcal{S}$ to be at most B with probability one. Similarly, in the FRMatCenter problem, we are given a matroid $\mathcal{M}$ and we require the solution $\mathcal{S}$ to be an independent set of $\mathcal{M}$ with probability one.

1.2 Our contributions and techniques

First of all, we give tight approximation algorithms for the RkCenter and RMatCenter problems.

▶ **Theorem 1.** *There exist a 2-approximation algorithm for the RkCenter problem* [1] *and a 3-approximation algorithm for the RMatCenter problem.*

Our main results for the lottery model are summarized in the following theorems.

▶ **Theorem 2.** *For any given constant $\epsilon > 0$ and any instance $\mathcal{I} = (V, d, k, t, \vec{p})$ of the FRkCenter problem, there is a randomized polynomial-time algorithm $\mathcal{A}$ which can compute a random solution $\mathcal{S}$ such that*

- *$|\mathcal{S}| \leq k$ with probability one,*
- *$|\mathcal{C}| \geq (1-\epsilon)t$, where $\mathcal{C}$ is the set of all clients within radius $2R$ from some center in $\mathcal{S}$ and R is the optimal radius,*
- *$\Pr[j \in \mathcal{C}] \geq (1-\epsilon)p_j$ for all $j \in V$.*

▶ **Theorem 3.** *For any $\epsilon > 0$ and any instance $\mathcal{I} = (V, d, w, B, t, \vec{p})$ of the FRKnapCenter problem, there is a randomized polynomial-time algorithm $\mathcal{A}$ which can return random solution $\mathcal{S}$ such that*

- *$\sum_{i \in \mathcal{S}} w_i \leq (1+\epsilon)B$ with probability one,*
- *$|\mathcal{C}| \geq t$, where $\mathcal{C}$ is the set of vertices within distance $3R$ from some vertex in $\mathcal{S}$,*
- *$\Pr[j \in \mathcal{C}] \geq p_j$ for all $j \in V$.*

Finally, the FRMatCenter can be reduced to (randomly) rounding a point in a matroid intersection polytope. We design a randomized rounding algorithm which can output a pseudo solution, which consists of a basis plus one extra center. By using a preprocessing step and a configuration LP, we can satisfy the matroid constraint exactly (respectively, knapsack constraint) while slightly violating the coverage and fairness constraints in the FRMatCenter (respectively, FRKnapCenter) problem. We believe these techniques could be useful in other facility-location problems (e.g., the matroid median problem [7, 10]) as well.

[1] A 2-approximation algorithm has also been found independently by Chakrabarty et. al. [1], and in a private discussion between Marek Cygan and Samir Khuller. Our algorithm here is different from the algorithm in [1].

▶ **Theorem 4.** *For any given constant $\gamma > 0$ and any instance $\mathcal{I} = (V, d, \mathcal{M}, t, \vec{p})$ of the FRMatCenter (respectively, FRKnapCenter) problem, there is a randomized polynomial-time algorithm $\mathcal{A}$ which can return a random solution $\mathcal{S}$ such that*

- *$\mathcal{S}$ is a basis of $\mathcal{M}$ with probability one, (respectively, $w(\mathcal{S}) \leq B$ with probability one)*
- *$|\mathcal{C}| \geq t - \gamma^2 n$, where $\mathcal{C}$ is the set of vertices within distance $3R$ from some vertex in $\mathcal{S}$,*
- *there exists a set $T \subseteq V$ of size at least $(1-\gamma)n$, which is deterministic, such that $\Pr[j \in \mathcal{C}] \geq p_j - \gamma$ for all $j \in T$.*

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we review some basic properties of matroids and discuss a filtering algorithm which is used in later algorithms. Then we develop approximation algorithms for the FRkCenter, FRKnapCenter, and FRMatCenter problems in the next three sections.

2 Preliminaries

2.1 Matroid polytopes

We first review a few basic facts about matroid polytopes. For any vector z and set S, we let $z(S)$ denote the sum $\sum_{i \in S} z_i$. Let $\mathcal{M}$ be any matroid on the ground set Ω and $r_{\mathcal{M}}$ be its rank function. The matroid base polytope of $\mathcal{M}$ is defined by

$$\mathcal{P}_{\mathcal{M}} := \left\{ x \in \mathbb{R}^{\Omega} : x(S) \leq r_{\mathcal{M}}(S) \;\; \forall S \subseteq \Omega; \quad x(\Omega) = r_{\mathcal{M}}(\Omega); \quad x_i \geq 0 \;\; \forall i \in \Omega \right\}.$$

▶ **Definition 5.** Suppose $Ax \leq b$ is a valid inequality of $\mathcal{P}_{\mathcal{M}}$. A face D of $\mathcal{P}_{\mathcal{M}}$ (corresponding to this valid inequality) is the set $D := \{x \in \mathcal{P}_{\mathcal{M}} : Ax = b\}$.

The following theorem gives a characterization for any face of $\mathcal{P}_{\mathcal{M}}$ (See, e.g., [9, 8]).

▶ **Theorem 6.** *Let D be any face of $\mathcal{P}_{\mathcal{M}}$. Then it can be characterized by*

$$D = \left\{ x \in \mathbb{R}^{\Omega} : x(S) = r_{\mathcal{M}}(S) \;\; \forall S \in \mathcal{L}; \quad x_i = 0 \;\; \forall i \in J; \quad x \in \mathcal{P}_{\mathcal{M}} \right\},$$

where $J \subseteq \Omega$ and $\mathcal{L}$ is a chain family of sets: $L_1 \subset L_2 \subset \ldots \subset L_m$. Moreover, it is sufficient to choose $\mathcal{L}$ as any maximal chain $L_1 \subset L_2 \subset \ldots \subset L_m$ such that $x(L_i) = r_{\mathcal{M}}(L_i)$ for all $i = 1, 2, \ldots, m$.

▶ Proposition 1. Let $x \in \mathcal{P}_{\mathcal{M}}$ be any point and I be the set of all tight constraints of $\mathcal{P}_{\mathcal{M}}$ on x. Suppose D is the face with respect to I. Then one can compute a chain family $\mathcal{L}$ for D as in Theorem 6 in polynomial time.

▶ **Corollary 7.** *Let D be any face of $\mathcal{P}_{\mathcal{M}}$. Then it can be characterized by*

$$D = \left\{ x \in \mathbb{R}^{\Omega} : x(S) = b_S \;\; \forall S \in \mathcal{O}; \quad x_i = 0 \;\; \forall i \in J; \quad x \in \mathcal{P}_{\mathcal{M}} \right\},$$

where $J \subseteq \Omega$ and $\mathcal{O}$ is a family of pairwise disjoint sets: $O_1, O_2, \ldots, O_m$, and $b_{O_1}, \ldots, b_{O_m}$ are some integer constants.

2.2 Filtering algorithm

All algorithms in this paper are based on rounding an LP solution. In general, for each vertex $i \in V$, we have a variable $y_i \in [0, 1]$ which represents the probability that we want to pick i in our solution. (In the standard model, y_i is the "extent" that i is opened.) In addition, for each pair of $i, j \in V$, we have a variable $x_{ij} \in [0, 1]$ which represents the probability that j is connected to i.

Note that in all center-type problems, the optimal radius R is always the distance between two vertices. Therefore, we can always "guess" the value of R in $O(n^2)$ time. WLOG, we may assume that we know the correct value of R. For any $j \in V$, we let $F_j := \{i \in V : d(i, j) \leq R \wedge x_{ij} > 0\}$ and $s_j := \sum_{i \in V : d(i,j) \leq R} x_{ij}$. We shall refer to F_j as a cluster with cluster center j. Depending on a specific problem, we may have different constraints on x_{ij}'s and y_i's. In general, the following constraints are valid in most of the problems here:

$$\sum_{j \in V} \sum_{i \in V : d(i,j) \leq R} x_{ij} \geq t, \tag{1}$$

$$\sum_{i \in V : d(i,j) \leq R} x_{ij} \leq 1, \quad \forall j \in V, \tag{2}$$

$$x_{ij} \leq y_i, \quad \forall i, j \in V, \tag{3}$$

$$y_i, x_{ij} \geq 0, \quad \forall i, j \in V. \tag{4}$$

For the *fair* variants, we may also require that

$$\sum_{i \in V : d(i,j) \leq R} x_{ij} \geq p_j, \quad \forall j \in V. \tag{5}$$

Constraint (1) says that at least t vertices should be covered. Constraint (2) ensures that each vertex is only connected to at most one center. Constraint (3) means vertex j can only connect to center i if it is open. Constraint (5) says that the total probability of j being connected should be at least p_j. By constraints (2) and (3), we have $y(F_j) \leq 1$.

The first step of all algorithms in this paper is to use the following *filtering* algorithm to obtain a maximal collection of disjoint clusters. The algorithm will return the set V' of cluster centers of the chosen clusters. In the process, we also keep track of the number c_j of other clusters removed by F_j for each $j \in V'$.

Algorithm 1 RFiltering (x, y)

1: $V' \leftarrow \emptyset$
2: **for each** cluster F_j in **decreasing order** of $s_j = \sum_{i \in V : d(i,j) \leq R} x_{ij}$ **do**
3:   **if** F_j is unmarked **then**
4:     $V' \leftarrow V' \cup \{j\}$
5:     Set all unmarked clusters F_k (including F_j itself) s.t. $F_k \cap F_j \neq \emptyset$ as marked.
6:     Let c_j be the number of marked clusters in this step.
7: $\vec{c} \leftarrow (c_j : j \in V')$
8: **return** $(V', \vec{c})$

3 The k-center problems with outliers

In this section, we first give a simple 2-approximation algorithm for the RkCenter problem. Then, we give an approximation algorithm for the FRkCenter problem, proving Theorem 2.

3.1 The robust k-center problem

Suppose $\mathcal{I} = (V, d, k, t)$ is an instance the RkCenter problem with the optimal radius R. Consider the polytope $\mathcal{P}_{\mathsf{RkCenter}}$ containing points (x, y) satisfying constraints (1)–(4), and the cardinality constraint:

$$\sum_{i \in V} y_i \leq k. \tag{6}$$

Since R is the optimal radius, it is not difficult to check that $\mathcal{P}_{\mathsf{RkCenter}} \neq \emptyset$. Let us pick any fractional solution $(x, y) \in \mathcal{P}_{\mathsf{RkCenter}}$. The next step is to round (x, y) into an integral solution using the following simple algorithm.

Algorithm 2 RkCenterRound (x, y)

1: $(V', \vec{c}) \leftarrow$ RFiltering (x, y).
2: $\mathcal{S} \leftarrow$ the top k vertices $i \in V'$ with highest value of c_i.
3: **return** $\mathcal{S}$

Analysis. By construction, the algorithm returns a set $\mathcal{S}$ of k open centers. Note that, for each $i \in \mathcal{S}$, c_i is the number of distinct clients within radius $2R$ from i. Thus, it suffices to show that $\sum_{i \in \mathcal{S}} c_i \geq t$. By inequality (2), we have that $s_j \leq 1$ for all $j \in V'$. Thus,

$$\sum_{i \in V'} c_i s_i \geq \sum_{i \in V} s_i \geq t,$$

where the first inequality is due to the greedy choice of vertices in V' and the second inequality follows by (1). Now recall that the clusters whose centers in V' are pairwise disjoint. By constraint (6), we have

$$\sum_{i \in V'} s_i \leq \sum_{i \in V'} y(F_i) \leq \sum_{i \in V} y_i \leq k.$$

It follows by the choice of $\mathcal{S}$ that $\sum_{i \in \mathcal{S}} c_i \geq t$. This concludes the first part of Theorem 1.

3.2 The fair robust k-center problem

Assume $\mathcal{I} = (V, d, k, t, \vec{p})$ be an instance of the FRkCenter problem with the optimal radius R. Fix any $\epsilon > 0$. If $k \leq 2/\epsilon$, then we can generate all possible $O\left(n^{1/\epsilon}\right)$ solutions and then solve an LP to obtain the corresponding marginal probabilities. So the problem can be solved easily in this case. We will assume that $k \geq 2/\epsilon$ for the rest of this section. Consider the polytope $\mathcal{P}_{\mathsf{FRkCenter}}$ containing points (x, y) satisfying constraints (1)–(4), the fairness constraint (5), and the cardinality constraint (6). We now show that $\mathcal{P}_{\mathsf{FRkCenter}}$ is actually a valid relaxation polytope.

► Proposition 2. We have that $\mathcal{P}_{\mathsf{FRkCenter}} \neq \emptyset$.

Fix any small parameter $\epsilon > 0$. The description of our algorithm is shown in Algorithm 3.

Analysis. First, note that one can find such a vector δ in line 5 as the system of $\delta(V') = 0$ and $\vec{c} \cdot \delta = 0$ consists of two constraints and at least 3 variables (and hence is underdetermined.) By construction, at least one more fractional variable becomes rounded after each iteration. Thus, the algorithm terminates after $O(n)$ rounds. Let $\mathcal{S}$ denote the (random) solution returned by FRkCenterRound and $\mathcal{C}$ be the set of all clients within radius $3R$ from some center in $\mathcal{S}$. Theorem 2 can be verified by the following propositions.

Algorithm 3 FRkCenterRound (ϵ, x, y)

1: $(V', \vec{c}) \leftarrow$ RFiltering (x, y) .
2: **for each** $j \in V'$ **do**
3: $y'_j \leftarrow (1-\epsilon)\sum_{i \in F_j} x_{ij}$
4: **while** y' still contains ≥ 3 fractional values in $(0,1)$ **do**
5: Let $\delta \in \mathbb{R}^{V'}, \delta \neq 0$ be such that $\delta_i = 0 \ \ \forall i \in V' : y'_i \in \{0,1\}$, $\delta(V') = 0$, and $\vec{c} \cdot \delta = 0$.
6: Choose scaling factors $a, b > 0$ such that
- $y' + a\delta \in [0,1]^{V'}$ and $y' - b\delta \in [0,1]^{V'}$
- there is at least one new entry of $y' + a\delta$ which is equal to zero or one
- there is at least one new entry of $y' - b\delta$ which is equal to zero or one

7: With probability $\frac{b}{a+b}$, update $y' \leftarrow y' + a\delta$; else, update $y' \leftarrow y' - b\delta$.
8: **return** $\mathcal{S} = \{i \in V : y'_i > 0\}$.

▶ Proposition 3. $|\mathcal{S}| \leq k$ with probability one.

▶ Proposition 4. $|\mathcal{C}| \geq (1-\epsilon)t$ with probability one.

▶ Proposition 5. $\Pr[j \in \mathcal{C}] \geq (1-\epsilon)p_j$ for all $j \in V$.

4 The Knapsack Center problems with outliers

We study the RKnapCenter and FRKnapCenter problems in this section. Recall that in these problems, each vertex has a weight and we want to make sure that the total weight of the chosen centers does not exceed a given budget B. We first give a 3-approximation algorithm for the RKnapCenter problem that slightly violates the knapsack constraint. Although this is not better than the known result by [3], both our algorithm and analysis here are more natural and simpler. It serves as a starting point for the next results. For the FRKnapCenter, we show that it is possible to satisfy the knapsack constraint exactly with small violations in the coverage and fairness constraints.

4.1 The robust knapsack center problem

Suppose $\mathcal{I} = (V, d, w, B, t)$ is an instance the RKnapCenter problem with the optimal radius R. Consider the polytope $\mathcal{P}_{\mathsf{RKnapCenter}}$ containing points (x, y) satisfying constraints (1)–(4), and the knapsack constraint:

$$\sum_{i \in V} w_i y_i \leq B. \tag{7}$$

Again, it is not difficult to check that $\mathcal{P}_{\mathsf{RKnapCenter}} \neq \emptyset$. Let us pick any fractional solution $(x, y) \in \mathcal{P}_{\mathsf{RKnapCenter}}$. See Algorithm 4 for the pseudo-approximation algorithm to round (x, y).

Analysis. We first claim that $\mathcal{P}' \neq \emptyset$ which implies that the extreme point Y of $\mathcal{P}'$ (in line 4) does exist. To see this, let $z_i := s_i$ for all $i \in V'$. Then we have

$$\sum_{i \in V'} c_i z_i = \sum_{i \in V'} c_i s_i \geq \sum_{i \in V} s_i \geq t.$$

Algorithm 4 RKnapCenterRound (x, y)

1: $(V', \vec{c}) \leftarrow$ RFiltering (x, y) .
2: For each $i \in V'$, let $v_i \leftarrow \arg\min_{j \in F_i}\{w_j\}$ be the vertex with smallest weight in F_i
3: Let $\mathcal{P}' := \left\{ z \in [0,1]^{V'} : \sum_{i \in V'} c_i z_i \geq t \quad \wedge \quad \sum_{i \in V'} w_{v_i} z_i \leq B \right\}$
4: Compute an extreme point Y of $\mathcal{P}'$
5: **return** $\mathcal{S} = \{v_i : i \in V,\ Y_i > 0\}$

Also,

$$\begin{aligned}\sum_{i \in V'} w_{v_i} z_i &= \sum_{i \in V'} w_{v_i} s_i \\ &= \sum_{i \in V'} w_{v_i} \sum_{j \in F_i} x_{ji} \\ &\leq \sum_{i \in V'} w_{v_i} \sum_{j \in F_i} y_j \\ &\leq \sum_{i \in V'} \sum_{j \in F_i} w_j y_j \leq \sum_{i \in V} w_i y_i \leq B.\end{aligned}$$

All the inequalities follow from LP constraints and definitions of s_i, c_i, and v_i. Thus, $z \in \mathcal{P}'$, implying that $\mathcal{P}' \neq \emptyset$.

▶ Proposition 6. RKnapCenterRound returns a solution $\mathcal{S}$ such that $w(\mathcal{S}) \leq B + 2w_{\max}$ and $|\mathcal{C}| \geq t$, where $\mathcal{C}$ is the set of vertices within distance $3R$ from some vertex in $\mathcal{S}$ and $w_{\max}$ is the maximum weight of any vertex in V.

4.2 The fair robust knapsack center problem

In this section, we will first consider a simple algorithm that only violates the knapsack constraint by two times the maximum weight of any vertex. Then using a configuration polytope to "condition" on the set of "big" vertices, we show that it is possible to either violate the budget by $(1 + \epsilon)$ or to preserve the knapsack constraint while slightly violating the coverage and fairness constraints.

4.2.1 Basic algorithm

Suppose $\mathcal{I} = (V, d, w, B, t, \vec{p})$ is an instance the FRKnapCenter problem with the optimal radius R. Consider the polytope $\mathcal{P}_{\mathsf{FRKnapCenter}}$ containing points (x, y) satisfying constraints (1)–(4), the fairness constraint (5), and the knapsack constraint (7). The proof that $\mathcal{P}_{\mathsf{FRKnapCenter}} \neq \emptyset$ is very similar to that of Proposition 2 and is omitted here.

The following algorithm is a randomized version of RKnapCenterRound.

Analysis. It is not hard to verify that $\mathcal{P}' \neq \emptyset$ (see the analysis in Section 4.1). This means that the decomposition at line 4 can be done.

▶ Proposition 7. The algorithm BasicFRKnapCenterRound returns a random solution $\mathcal{S}$ such that $w(\mathcal{S}) \leq B + 2w_{\max}$, $|\mathcal{C}| \geq t$, and $\Pr[j \in \mathcal{C}] \geq p_j$ for all $j \in V$, where $\mathcal{C}$ is the set of vertices within distance $3R$ from some vertex in $\mathcal{S}$ and $w_{\max}$ is the maximum weight of any vertex in V.

Algorithm 5 BasicFRKnapCenterRound (x, y)

1: $(V', \vec{c}) \leftarrow$ RFiltering (x, y) .
2: For each $i \in V'$ let $v_i := \arg\min_{j \in F_i}\{w_j\}$ be the vertex with smallest weight in F_i
3: Let $\mathcal{P}' := \left\{z \in [0,1]^{V'} : \sum_{i \in V'} c_i z_i \geq t \quad \wedge \quad \sum_{i \in V'} w_{v_i} z_i \leq B\right\}$
4: Let $z_i \leftarrow s_i$ for all $i \in V'$. Write z as a convex combination of extreme points $z^{(1)}, \ldots, z^{(n+1)}$ of $\mathcal{P}'$:

$$z = p_1 z^{(1)} + \ldots + p_{n+1} z^{(n+1)},$$

where $\sum_\ell p_\ell = 1$ and $p_\ell \geq 0$ for all $\ell \in [n+1]$.
5: Randomly choose $Y \leftarrow z_\ell$ with probability p_ℓ.
6: **return** $\mathcal{S} = \{v_i : i \in V,\ Y_i > 0\}$

4.2.2 An algorithm slightly violating the budget constraint

Fix a small parameter $\epsilon > 0$. A vertex i is said to be *big* iff $w_i > \epsilon B$. Then there can be at most $1/\epsilon$ big vertices in a solution. Let $\mathcal{U}$ denote the collection of all possible sets of big vertices. We have that $|\mathcal{U}| \leq n^{O(1/\epsilon)}$. Consider the *configuration* polytope $\mathcal{P}_{\mathsf{config1}}$ containing points (x, y, q) with the following constraints:

$$\begin{cases} \sum_{U \in \mathcal{U}} q_U = 1 & \\ \sum_{i \in V : d(i,j) \leq R} x_{ij}^U \leq q_U & \forall j \in V, U \in \mathcal{U} \\ \sum_{U \in \mathcal{U}} \sum_{i \in V : d(i,j) \leq R} x_{ij}^U \geq p_j & \forall j \in V \\ x_{ij}^U \leq y_i^U & \forall i, j \in V, U \in \mathcal{U} \\ \sum_{i \in V} w_i y_i^U \leq q_U B & \forall U \in \mathcal{U} \\ \sum_{j \in V} \sum_{i \in V : d(i,j) \leq R} x_{ij}^U \geq q_U t & \\ y_i^U = 1 & \forall U \in \mathcal{U}, i \in U \\ y_i^U = 0 & \forall U \in \mathcal{U}, i \in V \setminus U, w_i > 1/\epsilon \\ x_{ij}^U, y_i^U, q_U \geq 0 & \forall i, j \in V, U \in \mathcal{U} \end{cases}$$

We first claim that $\mathcal{P}_{\mathsf{config1}}$ is a valid relaxation polytope for the problem.

▶ Proposition 8. We have that $\mathcal{P}_{\mathsf{config1}} \neq \emptyset$.

Next, let us pick any $(x, y, q) \in \mathcal{P}_{\mathsf{config1}}$ and use the following algorithm to round it.

Algorithm 6 FRKnapCenterRound1 (x, y, q)

1: Randomly pick a set $U \in \mathcal{U}$ with probability q_U
2: Let $x'_{ij} \leftarrow x_{ij}^U / q_U$ and $y'_i \leftarrow \min\{y_i^U / q_U, 1\}$
3: **return** $\mathcal{S} =$ BasicRFKnapCenterRound (x', y')

We are now ready to prove Theorem 3.

Proof of Theorem 3. We will show that FRKnapCenterRound1 will return a solution $\mathcal{S}$ with properties in Theorem 3. Let $\mathcal{E}(U)$ denote the event that $U \in \mathcal{U}$ is picked in the algorithm. Note that (x', y') satisfies the following constraints:

$$\sum_{j \in V} \sum_{i \in V: d(i,j) \le R} x'_{ij} \ge t,$$
$$\sum_{i \in V: d(i,j) \le R} x'_{ij} \le 1, \quad \forall j \in V,$$
$$\sum_{i \in V: d(i,j) \le R} x'_{ij} = \sum_{i \in V: d(i,j) \le R} x_{ij}/q_U, \quad \forall j \in V,$$
$$x'_{ij} \le y'_i, \quad \forall i, j \in V,$$
$$\sum_{i \in V} w_i y'_i \le B.$$

Moreover, $y'_i = 1$ for all $i \in U$ and $y'_i = 0$ for all $i \in V \setminus U$ and $w_i > \epsilon B$. Thus, the two extra fractional vertices opened by BasicFRKnapCenterRound will have weight at most ϵB. By Proposition 7, we have $w(\mathcal{S}) \le B + 2\epsilon B = (1 + 2\epsilon)B$. Moreover, conditioned on U, we have

$$\Pr[j \in \mathcal{C} | \mathcal{E}(U)] \ge \sum_{i \in V: d(i,j) \le R} x'_{ij} = \sum_{i \in V: d(i,j) \le R} x_{ij}/q_U.$$

Thus, by definition of $\mathcal{P}_{\mathsf{config1}}$ and our construction of $\mathcal{S}$, we get

$$\begin{aligned}\Pr[j \in \mathcal{C}] &= \sum_{U \in \mathcal{U}} \Pr[j \in \mathcal{C} | \mathcal{E}(U)] \Pr[\mathcal{E}(U)] \\ &\ge \sum_{U \in \mathcal{U}} \sum_{i \in V: d(i,j) \le R} x_{ij} \\ &\ge p_j.\end{aligned}$$

◀

4.2.3 An algorithm that satisfies the knapsack constraint exactly

Let $\epsilon > 0$ a small parameter to be determined. Let $\mathcal{U}$ denote the collection of all possible sets of verticies with size at most $\lceil 1/\epsilon \rceil$. We have that $|\mathcal{U}| \le n^{O(1/\epsilon)}$. Suppose R is the optimal radius to our instance. Given a set $U \in \mathcal{U}$, we say that vertex $j \in V$ is *blue* if there exists $i \in U$ such that $d(i,j) \le 3R$. Otherwise, vertex i is said to be *red*. For any $i \in V$, let $\mathrm{RBall}(i, U, R)$ denote the set of red vertices within radius $3R$ from i:

$$\mathrm{RBall}(i, U, R) := \{ j \in V : (d(i,j) \le 3R \ \wedge \ \nexists k \in U : d(k,j) \le 3R) \}.$$

Consider the *configuration* polytope $\mathcal{P}_{\mathsf{config2}}$ containing points (x, y, q) with the following constraints:

$$\begin{cases} \sum_{U \in \mathcal{U}} q_U = 1 & \\ \sum_{i \in V: d(i,j) \le R} x^U_{ij} \le q_U & \forall j \in V, U \in \mathcal{U} \\ \sum_{U \in \mathcal{U}} \sum_{i \in V: d(i,j) \le R} x^U_{ij} \ge p_j & \forall j \in V \\ x^U_{ij} \le y^U_i & \forall i, j \in V, U \in \mathcal{U} \\ \sum_{i \in V} w_i y^U_i \le q_U B & \forall U \in \mathcal{U} \\ \sum_{j \in V} \sum_{i \in V: d(i,j) \le R} x^U_{ij} \ge q_U t & \\ y^U_i = 1 & \forall U \in \mathcal{U}, i \in U \\ y^U_i = 0 & \forall U \in \mathcal{U}, i \in V \setminus U, |\mathrm{RBall}(i, U, R)| \ge \epsilon n \\ x^U_{ij}, y^U_i, q_U \ge 0 & \forall i, j \in V, U \in \mathcal{U} \end{cases}$$

We first claim that $\mathcal{P}_{\mathsf{config2}}$ is a valid relaxation polytope for the problem.

▶ Proposition 9. We have that $\mathcal{P}_{\mathsf{config2}} \neq \emptyset$.

Next, let us pick any $(x, y, q) \in \mathcal{P}_{\mathsf{config2}}$ and use the Algorithm 7 to round it.

Algorithm 7 FRKNAPCENTERROUND2 (x, y, q)

1: Randomly pick a set $U \in \mathcal{U}$ with probability q_U
2: Let $x'_{ij} \leftarrow x^U_{ij}/q_U$ and $y'_i \leftarrow \min\{y^U_i/q_U, 1\}$
3: $\mathcal{S}' \leftarrow$ BASICRFKNAPCENTERROUND (x', y')
4: Let i_1, i_2 be vertices in $\mathcal{S}' \setminus U$ having largest weights.
5: **return** $\mathcal{S} = \mathcal{S}' \setminus \{i_1, i_2\}$

Analysis. Let us fix any $\gamma > 0$ and set $\epsilon := \frac{\gamma^2}{2}$. Also, let $\mathcal{E}(U)$ denote the event that $U \in \mathcal{U}$ is picked in the algorithm. Again, observe that (x', y') satisfies the following inequalities:

$$\sum_{j \in V} \sum_{i \in V : d(i,j) \le R} x'_{ij} \ge t,$$
$$\sum_{i \in V : d(i,j) \le R} x'_{ij} \le 1, \quad \forall j \in V,$$
$$\sum_{i \in V : d(i,j) \le R} x'_{ij} = \sum_{i \in V : d(i,j) \le R} x_{ij}/q_U, \quad \forall j \in V,$$
$$x'_{ij} \le y'_i, \quad \forall i, j \in V,$$
$$\sum_{i \in V} w_i y'_i \le B.$$

Recall that the algorithm BASICFRKNAPCENTERROUND will return a solution $\mathcal{S}'$ consisting of a set $\mathcal{S}''$ with $w(\mathcal{S}'') \le B$ plus (at most) two extra “fractional” centers i^* and i^{**}. Moreover, we have $0 < y'_{i^*}, y'_{i^{**}} < 0$, which implies that $i^*, i^{**} \notin U$. Thus, by removing the two centers having highest weights in $\mathcal{S}' \setminus U$, we ensure that the total weight of $\mathcal{S}$ is within the given budget B with probability one.

Now we shall prove the coverage guarantee. By Proposition 7, $\mathcal{S}'$ covers at least t vertices within radius $3R$. If a vertex is blue, it can always be connected to some center in U; and hence, it is not affected by the removal of i_1, i_2. Because each of i_1 and i_2 can cover at most ϵn other red vertices, we have

$$|\mathcal{C}| \ge t - 2\epsilon n = 1 - \gamma^2 n.$$

For any $j \in V$, let X_j be the random indicator for the event that j is covered by $\mathcal{S}'$ (i.e., there is some $i \in \mathcal{S}'$ such that $d(i, j) \le 3R$) but becomes unconnected due to the removal of i_1 or i_2. We say that j is a bad vertex iff $\mathrm{E}[X_j] \ge \gamma$. Otherwise, vertex j is said to be good. Note that $\sum_{j \in V} X_j \le 2\epsilon n$ with probability one. Thus, there can be at most $2\epsilon n/\gamma$ bad vertices. Let T be the set of all good vertices. Then

$$|T| \ge n - 2\epsilon n/\gamma = (1 - \gamma)n.$$

By Proposition 7, $\Pr[j \text{ is covered by } \mathcal{S}'] \geq p_j$. For any $j \in T$, we have

$$\begin{aligned}\Pr[j \in \mathcal{C}] &= \Pr[j \text{ is covered by } \mathcal{S}' \wedge X_j = 0] \\ &= \Pr[j \text{ is covered by } \mathcal{S}'] - \Pr[j \text{ is covered by } \mathcal{S}' \wedge X_j = 1] \\ &\geq \Pr[j \text{ is covered by } \mathcal{S}'] - \Pr[X_j = 1] \\ &\geq p_j - \gamma.\end{aligned}$$

This concludes the first part of Theorem 4 for the FRKnapCenter problem.

5 The Matroid Center problems with outliers

In this section, we will first give a tight 3-approximation algorithm for the RMatCenter problem, improving upon the 7-approximation algorithm by Chen et. al. [3]. Then we study the FRMatCenter problem and give a proof for the second part of Theorem 4.

5.1 The robust matroid center problem

Suppose $\mathcal{I} = (V, d, \mathcal{M}, t)$ is an instance the RMatCenter problem with the optimal radius R. Let $r_{\mathcal{M}}$ denote the rank function of $\mathcal{M}$. Consider the polytope $\mathcal{P}_{\mathsf{RMatCenter}}$ containing points (x, y) satisfying constraints (1)–(4), and the matroid rank constraints:

$$y(U) \leq r_{\mathcal{M}}(U), \quad \forall U \subseteq V. \tag{8}$$

Since R is the optimal radius, it is not difficult to check that $\mathcal{P}_{\mathsf{RMatCenter}} \neq \emptyset$. Let us pick any fractional solution $(x, y) \in \mathcal{P}_{\mathsf{RMatCenter}}$. The next step is to round (x, y) into an integral solution. Our 3-approximation algorithm is summarized in Algorithm 8.

Algorithm 8 RMatCenterRound (x, y)

1: $(V', \vec{c}) \leftarrow$ RFiltering (x, y) .
2: Let $\mathcal{P}' := \left\{z \in [0,1]^V : z(U) \leq r_{\mathcal{M}}(U)\ \forall U \subseteq V\ \wedge\ z(F_i) \leq 1\ \ \forall i \in V'\right\}$
3: Find a basic solution $Y \in \mathcal{P}'$ which maximizes the linear function $f : [0,1]^V \to \mathbb{R}$ defined as

$$f(z) := \sum_{j \in V'} c_j \sum_{i \in F_j} z_i \quad \text{for } z \in [0,1]^V.$$

4: **return** $\mathcal{S} = \{i \in V : Y_i = 1\}$.

Analysis. Again, by construction, the clusters F_i are pairwise disjoint for $i \in V'$. Note that $\mathcal{P}'$ is the matroid intersection polytope between $\mathcal{M}$ and another partition matroid polytope saying that at most one item per set F_i for $i \in V'$ can be chosen. Moreover, $y \in \mathcal{P}'$ implies that $\mathcal{P}' \neq \emptyset$. Thus, $\mathcal{P}'$ has integral extreme points and optimizing over $\mathcal{P}'$ can be done in polynomial time. Note that the solution $\mathcal{S}$ is feasible as it satisfies the matroid constraint. The correctness of RMatCenterRound follows immediately by the following two propositions.

▶ Proposition 10. There are at least $f(Y)$ vertices in V that are at distance at most $3R$ from some open center in $\mathcal{S}$.

▶ Proposition 11. We have that $f(Y) \geq t$.

This analysis proves the second part of Theorem 1.

5.2 The fair robust matroid center problem

In this section, we consider the FRMatCenter problem. It is not difficult to modify and randomize algorithm RMCenterRound so that it would return a random solution satisfying both the fairness guarantee and matroid constraint, and preserving the coverage constraint *in expectation*. This can be done by randomly picking Y inside $\mathcal{P}'$. However, if we want to obtain some concrete guarantee on the coverage constraint, we may have to (slightly) violate either the matroid constraint or the fairness guarantee. We leave it as an open question whether there exists a true approximation algorithm for this problem.

We will start with a pseudo-approximation algorithm which always returns a basis of $\mathcal{M}$ plus at most one extra center. Our algorithm is quite involved. We first carefully round a fractional solution inside a matroid intersection polytope into a (random) point with a special property: the unrounded variables form a single path connecting some clusters and tight matroid rank constraints. Next, rounding this point will ensure that all but one cluster have an open center. Then opening one extra center is sufficient to cover at least t clients.

Finally, using a similar preprocessing step similar to the one in Section 4.2.3, we can correct the solution by removing the extra center without affecting the fairness and coverage guarantees by too much. This algorithm concludes Theorem 4.

5.2.1 A pseudo-approximation algorithm

Suppose $\mathcal{I} = (V, d, \mathcal{M}, t, \vec{p})$ is an instance the robust matroid center problem with the optimal radius R. Let $r_{\mathcal{M}}$ denote the rank function of $\mathcal{M}$ and $\mathcal{P}_{\mathcal{M}}$ be the matroid base polytope of $\mathcal{M}$. Consider the polytope $\mathcal{P}_{\mathsf{FRMatCenter}}$ containing points (x, y) satisfying constraints (1)–(4), the fairness constraint (5), and the matroid constraints (8). Using similar arguments as in the proof of Proposition 2, we can show that $\mathcal{P}_{\mathsf{FRMatCenter}}$ is a valid relaxation.

▶ Proposition 12. We have that $\mathcal{P}_{\mathsf{FRMatCenter}} \neq \emptyset$.

Our algorithm will use the following rounding operation iteratively.

Algorithm 9 RoundSinglePoint $(y, \vec{r})$

1: $\delta^* \leftarrow \max\{\delta : z \in \mathcal{P}_{\mathcal{M}}; z_v = y_v + \delta r_v \ \forall v \in V\}$
2: $y' \leftarrow y + \delta^* \vec{r}$
3: **return** (y', δ^*)

Given a point $y \in \mathcal{P}_{\mathcal{M}}$ and a vector $\vec{r}$, the procedure RoundSinglePoint will move y along direction $\vec{r}$ to a new point $y + \delta^* \vec{r}$ for some maximal $\delta^* > 0$ such that this point still lies in $\mathcal{P}_{\mathcal{M}}$. Note that one can find such a maximal δ^* in polynomial time. We will choose the initial point (x, y) as a vertex of $\mathcal{P}_{\mathsf{FRMatCenter}}$. By Cramer's rule, the entries of y will be rational with both numerators and denominators bounded by $O(2^n)$. The direction vector $\vec{r}$ also has this property by construction. Thus, it is not hard to verify that the maximal value of δ^* for which $y + \delta^* \vec{r} \in \mathcal{P}_{\mathcal{M}}$ is also rational and has both numerator and denominator at most $O(2^n)$ in every iteration. So we can compute δ^* exactly by a simple binary search.

See the appendix for more details.

5.2.2 Analysis of PseudoFRMCenterRound

▶ Proposition 13. In all but the last iteration, the while-loop (lines 4 to 8) of PseudoFRMCenterRound preserves the following invariant: if y' lies in the face D of $\mathcal{P}_{\mathcal{M}}$ (w.r.t all tight matroid rank constraints) at the beginning of the iteration, then $y' \in D$ at the end of this iteration.

▶ Proposition 14. PseudoFRMCenterRound terminates in polynomial time.

▶ Proposition 15. In all iterations, the while-loop (lines 4 to 8) of PseudoFRMCenterRound satisfies the invariant that $y'(F_j) \leq 1$ for all $F_j \in \mathcal{F}$.

▶ Proposition 16. PseudoFRMCenterRound returns a solution $\mathcal{S}$ which is some independent set of $\mathcal{M}$ plus (at most) one extra vertex in V.

Recall that $\mathcal{C}$ is the (random) set of all clients within radius $3R$ from some center in $\mathcal{S}$, where R is the optimal radius. The following two propositions will conclude our analysis.

▶ Proposition 17. $|\mathcal{C}| \geq t$ with probability one.

▶ Proposition 18. $\Pr[j \in \mathcal{C}] \geq p_j$ for all $j \in V$.

So far we have proved the following theorem.

▶ **Theorem 8.** PseudoFRMCenterRound *will return a random solution $\mathcal{S}$ such that*
- *$\mathcal{S}$ is the union of some basis of $\mathcal{M}$ with (at most) one extra vertex,*
- *$|\mathcal{C}| \geq t$ with probability one,*
- *$\Pr[j \in \mathcal{C}] \geq p_j$ for all $j \in V$.*

5.2.3 An algorithm satisfying the matroid constraint exactly

Using a similar technique as in Section 4.2.3, we will develop an approximation algorithm for the FRMatCenter problem which always returns a feasible solution. Let $\epsilon > 0$ a small parameter to be determined. Let $\mathcal{U}$ denote the collection of all possible sets of verticies with size at most $\lceil 1/\epsilon \rceil$ such that U is an independent set of $\mathcal{M}$. Again, we have that $|\mathcal{U}| \leq n^{O(1/\epsilon)}$. Suppose R is the optimal radius to our instance. For any $i \in V$, recall that $\mathrm{RBall}(i, U, R)$ is the set of red vertices within radius $3R$ from i.

Consider the *configuration* polytope $\mathcal{P}_{\mathsf{config3}}$ containing points (x, y, q) with the following constraints:

$$\begin{cases}
\sum_{U \in \mathcal{U}} q_U = 1 & \\
\sum_{i \in V: d(i,j) \leq R} x_{ij}^U \leq q_U & \forall j \in V, U \in \mathcal{U} \\
\sum_{U \in \mathcal{U}} \sum_{i \in V: d(i,j) \leq R} x_{ij}^U \geq p_j & \forall j \in V \\
x_{ij}^U \leq y_i^U & \forall i, j \in V, U \in \mathcal{U} \\
\sum_{i \in W} y_i^U \leq q_U r_{\mathcal{M}}(W) & \forall U \in \mathcal{U}, W \subseteq V \\
\sum_{j \in V} \sum_{i \in V: d(i,j) \leq R} x_{ij}^U \geq q_U t & \\
y_i^U = 1 & \forall U \in \mathcal{U}, i \in U \\
y_i^U = 0 & \forall U \in \mathcal{U}, i \in V \setminus U, |\mathrm{RBall}(i, U, R)| \geq \epsilon n \\
x_{ij}^U, y_i^U, q_U \geq 0 & \forall i, j \in V, U \in \mathcal{U}
\end{cases}$$

We first claim that $\mathcal{P}_{\mathsf{config3}}$ is a valid relaxation polytope for the problem.

▶ Proposition 19. We have that $\mathcal{P}_{\mathsf{config3}} \neq \emptyset$.

Next, let us pick any $(x, y, q) \in \mathcal{P}_{\mathsf{config3}}$ and use Algorithm 10 to round it.

Algorithm 10 FRMCenterRound (x, y, q)

Randomly pick a set $U \in \mathcal{U}$ with probability q_U
Let $x'_{ij} \leftarrow x^U_{ij}/q_U$ and $y'_i \leftarrow \min\{y^U_i/q_U, 1\}$
$\mathcal{S}' \leftarrow$ PseudoFRMCenterRound (x', y')
Let i^* be the "extra" vertex in $\mathcal{S}'$.
return $\mathcal{S} = \mathcal{S}' \setminus \{i\}$

Analysis. We are now ready to prove the second part of Theorem 4. Let us fix any $\gamma > 0$ and set $\epsilon := \gamma^2$. Also, let $\mathcal{E}(U)$ denote the event that $U \in \mathcal{U}$ is picked in the algorithm. Note that (x', y') satisfies the following inequalities:

$$\sum_{j \in V} \sum_{i \in V : d(i,j) \le R} x'_{ij} \ge t,$$
$$\sum_{i \in V : d(i,j) \le R} x'_{ij} \le 1, \quad \forall j \in V,$$
$$\sum_{i \in V : d(i,j) \le R} x'_{ij} = \sum_{i \in V : d(i,j) \le R} x_{ij}/q_U, \quad \forall j \in V,$$
$$x'_{ij} \le y'_i, \quad \forall i, j \in V,$$
$$\sum_{i \in W} y'_i \le r_{\mathcal{M}}(W), \quad \forall W \subseteq V.$$

Moreover, $y'_i = 1$ for all $i \in U$ and $y'_i = 0$ for all $i \in V \setminus U$ and $\mathrm{RBall}(i, U, R) \ge \epsilon n$.

Recall that the algorithm PseudoFRMCenterRound will return a solution $\mathcal{S}'$ is the union of a basis of $\mathcal{M}$ with an extra center i^*. Moreover, we have $0 < y'_{i^*} < 0$, which implies that $i^* \notin U$. Thus, by removing i^* from $\mathcal{S}'$, we ensure that the resulting set is a basis of $\mathcal{M}$ with probability one.

Now we shall prove the coverage guarantee. By Theorem 8, $\mathcal{S}'$ covers at least t vertices within radius $3R$. If a vertex is blue, it can always be connected to some center in U; and hence, it is not affected by the removal of i_1, i_2. Because each of i^* can cover at most ϵn other red vertices, we have

$$|\mathcal{C}| \ge t - \epsilon n = 1 - \gamma^2 n.$$

For any $j \in V$, let X_j be the random indicator for the event that j is covered by $\mathcal{S}'$ (i.e., there is some $i \in \mathcal{S}'$ such that $d(i,j) \le 3R$) but becomes unconnected due to the removal of i^*. We say that j is a bad vertex iff $\mathrm{E}[X_j] \ge \gamma$. Otherwise, vertex j is said to be good. Again, $\sum_{j \in V} X_j \le \epsilon n$ with probability one. Thus, there can be at most $\epsilon n/\gamma$ bad vertices. Let T be the set of all good vertices. Then

$$|T| \ge n - \epsilon n/\gamma = (1 - \gamma) n.$$

By Theorem 8, $\Pr[j \text{ is covered by } \mathcal{S}'] \ge p_j$. So, for any $j \in T$, we have

$$\Pr[j \in \mathcal{C}] \ge \Pr[j \text{ is covered by } \mathcal{S}'] - \Pr[X_j = 1] \ge p_j - \gamma.$$

Acknowledgements. We thank the APPROX 2017 referees for their helpful suggestions.

A Appendix

A.1 Details of the pseudo-approximation for FRMatCenter

The main algorithm is summarized in Algorithm 11, which can round any *vertex* point $(x, y) \in \mathcal{P}_{\mathsf{FRMatCenter}}$. Basically, we will round y iteratively. In each round, we construct a (multi)-bipartite graph where vertices on the left side are the disjoint sets $O_1, O_2, \ldots$ in Corollary 7. Vertices on the right side are corresponding to the disjoint sets $F_1, F_2, \ldots$ returned by RFiltering. Now each edge of the bipartite graph, connecting O_i and F_j, represents some unrounded variable $y_v \in (0, 1)$ where $v \in O_i$ and $v \in F_j$. See Figure 1.

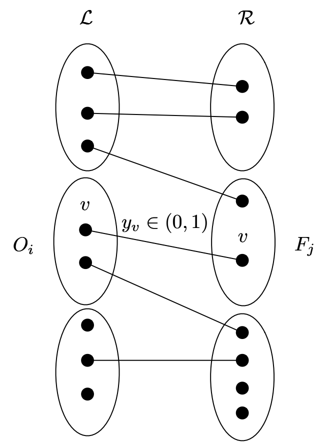

Figure 1 Construction of the multi-bipartite graph $H = (\mathcal{L}, \mathcal{R}, E_H)$ in the main algorithm.

Then we carefully pick a cycle (path) on this graph and round variables on the edges of this cycle (path). This is done by subroutines RoundCycle, RoundSinglePath, and RoundTwoPaths. See Figures 2, 3, and 4. Basically, these procedures will first choose a direction $\vec{r}$ which alternatively increases and decreases the variables on the cycle (path) so that (i) all tight matroid constraints are preserved and (ii) the number of (fractionally) covered clients is also preserved. Now we randomly move y along $\vec{r}$ or $-\vec{r}$ using procedure RoundSinglePoint to ensure that all the marginal probabilities are preserved.

Finally, all the remaining, fractional variables will form one path on the bipartite graph. We round these variables by the procedure RoundFinalPath which exploits the integrality of any face of a matroid intersection polytope. Then, to cover at least t clients, we may need to open one extra facility.

Algorithm 11 PSEUDOFRMCENTERROUND (x, y)

1: $(V', \vec{c}) \leftarrow$ RFILTERING (x, y) and let $\mathcal{F} \leftarrow \{F_j : j \in V'\}$
2: Set $y'_i \leftarrow x_{ij}$ for all $j \in V', i \in F_j$
3: Set $y'_i \leftarrow 0$ for all $i \in V \setminus \bigcup_{j \in V'} F_j$
4: **while** y' still contains some fractional values **do**
5: Note that $y' \in \mathcal{P}_{\mathcal{M}}$. Compute the disjoint sets $O_1, \ldots, O_t$ and constants $b_{O_1}, \ldots, b_{O_t}$ as in Corollary 7.
6: Let $O_0 \leftarrow V \setminus \bigcup_{i=1}^{t} O_i$ and $F_0 \leftarrow V \setminus \bigcup_{j \in V'} F_j$
7: Construct a multi-bipartite graph $H = (\mathcal{L}, \mathcal{R}, E_H)$ where
- each vertex $i \in \mathcal{L}$, where $\mathcal{L} = \{0, \ldots, t\}$, is corresponding to the set O_i
- each vertex $j \in \mathcal{R}$, where $\mathcal{R} = \{0\} \cup \{k : F_k \in \mathcal{F}\}$, is corresponding to the set F_j
- for each vertex $v \in V$ such that $y_v \in (0, 1)$: if v belongs to some set O_i and F_j, add an edge e with label v connecting $i \in \mathcal{L}$ and $j \in \mathcal{R}$.

8: Check the following cases (in order):
- Case 1: H contains a cycle. Let $\vec{v} = (v_1, v_2, \ldots, v_{2\ell})$ be the sequence of edge labels on this cycle. Update $y' \leftarrow$ ROUNDCYCLE$(y', \vec{v})$ and go to line 4.
- Case 2: H contains a maximal path with one endpoint in $\mathcal{L}$ and the other in $\mathcal{R}$. Let $\vec{v} = (v_1, v_2, \ldots, v_{2\ell+1})$ be the sequence of edge labels on this path. Update $y' \leftarrow$ ROUNDSINGLEPATH$(y', \vec{v})$ and go to line 4.
- Case 3: There are at least 2 distinct maximal paths (not necessarily disjoint) having both endpoints in $\mathcal{R}$. Let $\vec{v}_1, \vec{v}_2$ be the sequences of edge labels on these two paths. Update $y' \leftarrow$ ROUNDTWOPATHS$(y', \vec{v}_1, \vec{v}_2, \vec{c})$ and go to line 4.
- The remaining case: all edges in H form a single path with both endpoints in $\mathcal{R}$. Let $(v_1, v_2, \ldots, v_{2\ell})$ be the sequence of edge labels on this path. Let $Y \leftarrow$ ROUNDFINALPATH$(y', \vec{v})$ and exit the loop.

9: **return** $\mathcal{S} = \{i \in V : Y_i = 1\}$.

Algorithm 12 ROUNDCYCLE $(y', \vec{v})$

1: Initialize $\vec{r} = \vec{0}$, then set $r_{v_j} = (-1)^j$ for $j = 1, 2, \ldots, |\vec{v}|$
2: $(y_1, \delta_1) \leftarrow$ROUNDSINGLEPOINT$(y', \vec{r})$
3: **return** y_1

Algorithm 13 ROUNDSINGLEPATH $(y', \vec{v})$

1: Initialize $\vec{r} = \vec{0}$, then set $r_{v_j} = (-1)^{j+1}$ for $j = 1, 2, \ldots, |\vec{v}|$
2: $(y_1, \delta_1) \leftarrow$ROUNDSINGLEPOINT$(y', \vec{r})$
3: **return** y_1

Algorithm 14 ROUNDTWOPATHS $(y', \vec{v}, \vec{v}', \vec{c})$

1: WLOG, suppose $j_1, j_2 \in \mathcal{R}$ are endpoints of $v_1, v_{2\ell}$ of the path $\vec{v}$ respectively and $c_{j_1} \geq c_{j_2}$

2: WLOG, suppose $j'_1, j'_2 \in \mathcal{R}$ are endpoints of $v'_1, v'_{2\ell'}$ of the path $\vec{v}'$ respectively and $c_{j'_1} \geq c_{j'_2}$

3: $\Delta_1 \leftarrow c_{j_1} - c_{j_2}$; $\Delta_2 \leftarrow c_{j'_1} - c_{j'_2}$; $\vec{r} \leftarrow \vec{0}$

4: $V_1^+ \leftarrow \{v_1, v_3, \ldots, v_{2\ell-1}\}$; $V_1^- \leftarrow \{v_2, v_4, \ldots, v_{2\ell}\}$

5: $V_2^+ \leftarrow \{v'_2, v'_4, \ldots, v'_{2\ell'}\}$; $V_2^- \leftarrow \{v'_1, v'_3, \ldots, v'_{2\ell'-1}\}$

6: **for each** $v \in V_1^+$: $r_v \leftarrow r_v + 1$; **for each** $v \in V_1^-$: $r_v \leftarrow r_v - 1$

7: **for each** $v \in V_2^+$: $r_v \leftarrow r_v + \Delta_1/\Delta_2$; **for each** $v \in V_2^-$: $r_v \leftarrow r_v - \Delta_1/\Delta_2$

8: $(y_1, \delta_1) \leftarrow$ROUNDSINGLEPOINT$(y', \vec{r})$

9: $(y_2, \delta_2) \leftarrow$ROUNDSINGLEPOINT$(y', -\vec{r})$

10: With probability $\delta_1/(\delta_1 + \delta_2)$: **return** y_2

11: With remaining probability $\delta_2/(\delta_1 + \delta_2)$: **return** y_1

Algorithm 15 ROUNDFINALPATH $(y, \vec{v})$

1: $\mathcal{P}_1 \leftarrow \left\{z \in [0,1]^V : z(U) \leq r_{\mathcal{M}}(U)\ \forall U \subseteq V \wedge z(O_i) = b_{O_i}\ \forall i \in \mathcal{L} \setminus \{0\} \wedge z_i = 0\ \forall i : y_i = 0\right\}$

2: $\mathcal{P}_2 \leftarrow \{z \in [0,1]^V : z(F_j) = y(F_j)\ \forall j \in V' \setminus J \wedge z(F_j) \leq 1\ \forall j \in J\}$, where $J \subseteq \mathcal{R}$ is the set of vertices in $\mathcal{R}$ on the path $\vec{v}$.

3: Pick an arbitrary extreme point $\hat{y}$ of $\mathcal{P}' = \mathcal{P}_1 \cap \mathcal{P}_2$

4: **for each** $j \in \mathcal{R}$ and j is on the path $\vec{v}$: if $\hat{y}(F_j) = 0$, pick an arbitrary $u \in F_j$ and set $\hat{y}_u \leftarrow 1$.

5: **return** $\hat{y}$

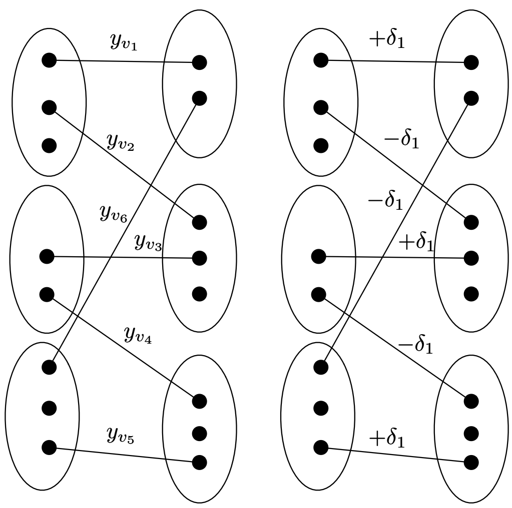

Figure 2 The left part shows a cycle. The right part shows how the variables on the cycle are being changed by ROUNDCYCLE.

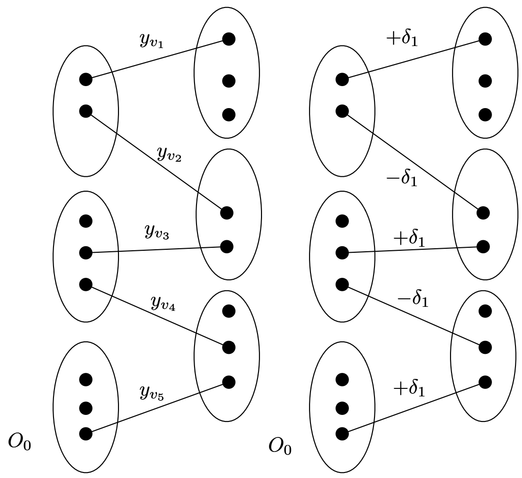

Figure 3 The left part shows a single path. The right part shows how the variables on the path are being changed by RoundSinglePath.

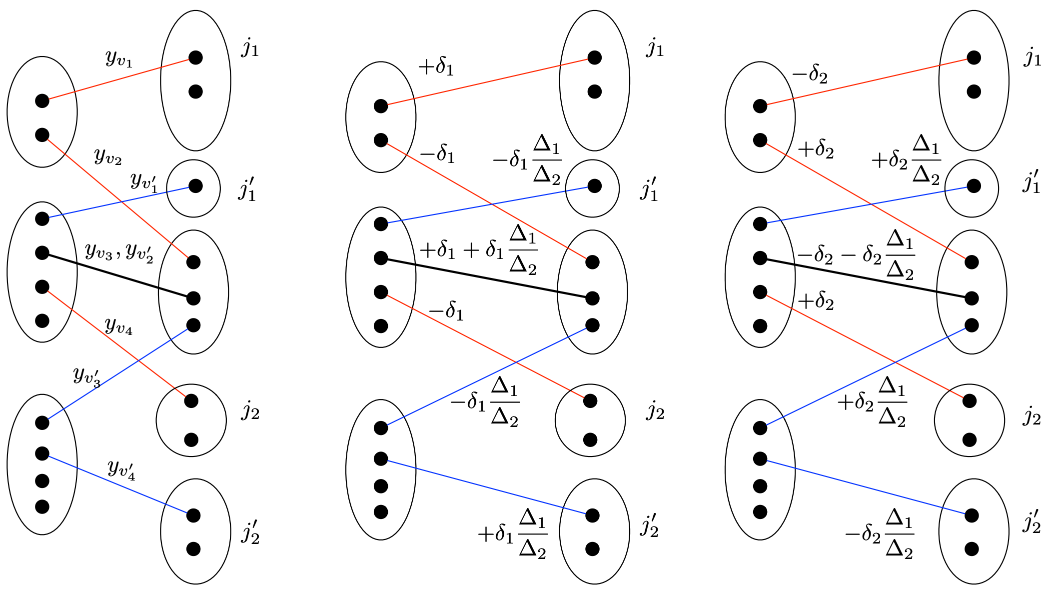

Figure 4 The left part shows an example of two distinct maximal paths chosen in Case 3. The black edge is common in both paths. The middle and right parts are two possibilities of rounding y. With probability $\delta_1/(\delta_1 + \delta_2)$, the strategy in the right part is adopted. Otherwise, the strategy in the middle part is chosen.

References

1 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The non-uniform k-center problem. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, pages 67:1–67:15, 2016.

2 Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 642–651, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=365411.365555.

3 Danny Z. Chen, Jian Li, Hongyu Liang, and Haitao Wang. Matroid and knapsack center problems. *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, pages 110–122, 2013. URL: http://dx.doi.org/10.1007/978-3-642-36694-9_10, doi:10.1007/978-3-642-36694-9_10.

4 David Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. Fairness in resource allocation and slowed-down dependent rounding. Manuscript, 2017.

5 Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, May 1986. URL: http://doi.acm.org/10.1145/5925.5933, doi:10.1145/5925.5933.

6 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209 – 215, 1979. URL: http://www.sciencedirect.com/science/article/pii/0166218X79900441, doi:http://dx.doi.org/10.1016/0166-218X(79)90044-1.

7 Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. In *Proceedings of the annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1117–1130. SIAM, 2011.

8 Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization.* Cambridge University Press, New York, NY, USA, 1st edition, 2011.

9 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

10 Chaitanya Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. In *APPROX/RANDOM 2014*, volume 28, pages 403–418, 2014.