Bicriteria Approximation Algorithms for Priority Matroid Median

Tanvi Bajpai ✉
Department of Computer Science, University of Illinois, Urbana-Champaign, Urbana, IL, USA

Chandra Chekuri ✉
Department of Computer Science, University of Illinois, Urbana-Champaign, Urbana, IL, USA

Abstract

Fairness considerations have motivated new clustering problems and algorithms in recent years. In this paper we consider the Priority Matroid Median problem which generalizes the Priority k-Median problem that has recently been studied. The input consists of a set of facilities $\mathcal{F}$ and a set of clients $\mathcal{C}$ that lie in a metric space $(\mathcal{F} \cup \mathcal{C}, d)$, and a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$ over the facilities. In addition, each client j has a specified radius $r_j \geq 0$ and each facility $i \in \mathcal{F}$ has an opening cost $f_i > 0$. The goal is to choose a subset $S \subseteq \mathcal{F}$ of facilities to minimize $\sum_{i \in \mathcal{F}} f_i + \sum_{j \in \mathcal{C}} d(j, S)$ subject to two constraints: (i) S is an independent set in $\mathcal{M}$ (that is $S \in \mathcal{I}$) and (ii) for each client j, its distance to an open facility is at most r_j (that is, $d(j, S) \leq r_j$). For this problem we describe the first bicriteria (c_1, c_2) approximations for fixed constants c_1, c_2: the radius constraints of the clients are violated by at most a factor of c_1 and the objective cost is at most c_2 times the optimum cost. We also improve the previously known bicriteria approximation for the uniform radius setting ($r_j := L \ \forall j \in \mathcal{C}$).

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

Keywords and phrases k-median, fair clustering, matroid

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.7

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2210.01888`

Funding *Tanvi Bajpai*: Supported in part by NSF grant CCF-1910149.
Chandra Chekuri: Supported in part by NSF grants CCF-1910149 and CCF-1907937.

1 Introduction

Clustering and facility-location problems are widely studied in areas such as machine learning, operations research, and algorithm design. Among these, center-based clustering problems in metric spaces form a central topic and will be our focus. The input for these problems is a set of clients $\mathcal{C}$ and a set of facilities $\mathcal{F}$ from a metric space $(\mathcal{F} \cup \mathcal{C}, d)$. The goal is to select a subset of facilities $S \subseteq \mathcal{F}$ to open, subject to various constraints, so as to minimize an objective that depends on the distances of the clients to the chosen centers; we use $d(j, S)$ to denote the quantity $\min_{i \in S} d(j, i)$ which is the distance from j to S. Typical objectives are of the form $(\sum_{j \in \mathcal{C}} d(j, S)^p)^{1/p}$ for some parameter p (the ℓ_p norm of the distances). When the constraint on facilities is that at most k can be chosen (that is, $|S| \leq k$), we obtain several standard and well-studied problems such as k-Center ($p = \infty$), k-Median ($p = 1$), and k-Means ($p = 2$) problems. These problems are extensively studied from many perspectives [15, 25, 10, 2, 5, 19, 16]. These are also well-studied in the geometric setting when $\mathcal{F}$ is the continuous space $\mathbb{R}^\ell$ for some finite dimension ℓ. In this paper we restrict our attention to the discrete setting, and in particular, to the median objective ($p = 1$).

The Matroid Median problem is a generalization of the k-Median clustering problem. Here, the cardinality constraint k on S is replaced by specifying a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$ on the facility set $\mathcal{F}$ and requiring that $S \in \mathcal{I}$ (we refer a reader unfamiliar with matroids to

Section 2 formal definitions and details). The k-Median clustering problem can be written as an instance of Matroid Median where $\mathcal{M}$ is the uniform matroid of rank k. The Matroid Median problem was first introduced by Krishnaswamy et al. [21] as a generalization of k-Median and Red-Blue Median [14].

Motivated by the versatility of the Matroid Median problem, and several other considerations that we will discuss shortly, in this paper we study the Priority Matroid Median problem (PMatMed). Formally, in PMatMed we are given a set of clients $\mathcal{C}$ and facilities $\mathcal{F}$ from a metric space $(\mathcal{F} \cup \mathcal{C}, d)$ where each facility $i \in \mathcal{F}$ has a facility opening cost f_i, and each client $j \in \mathcal{C}$ has a radius value r_j. We are also given a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$ over the facilities. The goal is to select a subset of facilities S that is an independent set of the matroid $\mathcal{M}$ where the objective $\sum_{j \in \mathcal{C}} d(j, S) + \sum_{i \in S} f_i$ (i.e. the *cost* induced by selected facilities) is minimized, and the *radius* constraint $d(j, S) \leq r_j$ is satisfied for all clients $j \in \mathcal{C}$.

Most of the center-based clustering problems are NP-Hard even in very restricted settings. We focus on polynomial-time approximation algorithms which have an extensive history in center-based clustering. Moreover, due to the nature of the constraints in PMatMed, we can only obtain bicriteria approximation guarantees that violate both the objective and the radius constraints. An (α, β)-approximation algorithm for PMatMed is a polynomial-time algorithm that either correctly states that no feasible solution is possible or outputs a set of facilities $S \in \mathcal{I}$ (hence satisfies the matroid constraint) such that (i) $d(j, S) \leq \alpha r_j$ for all clients $j \in \mathcal{C}$ and (ii) the cost objective value of S is at most $\beta \cdot OPT$ where OPT is the cost of an optimum solution.

1.1 Motivation, Applications to Fair Clustering, and Related Work

Our study of PMatMed is motivated, at a high-level, by two considerations. First, there has been past work that combines the k-Median objective with that of the k-Center objective. Alamdari and Shmoys [3] considered the k-Median problem with the additional constraint that each client is served within a given uniform radius L and obtained a $(4, 8)$-approximation. Their work is partially motivated by the ordered median problem [24, 4, 8]. Kamiyama [18] studied a generalization of this uniform radius requirement on clients to the setting of Matroid Median and derived a $(11, 16)$-approximation algorithm. Note that this is a special case of PMatMed where $r_j = L$ for each j. We call this the UniPMatMed problem.

Another motivation for studying PMatMed is the recent interest in *fair clustering* in the broader context of algorithmic fairness. The goal is to capture and address social concerns in applications that rely on clustering procedures and algorithms. Various notions of fair clustering have been proposed. Chierchetti et al. [11] formulated the Fair k-Center problem: clients belong to one or more groups based on various attributes. The objective is to return a clustering of points where each chosen center services a representative number of clients from *every* group. This notion has since been classified as one that seeks to achieve *group* fairness. Several other group fair clustering problems have since been introduced and studied [7, 20, 1, 13]. Subsequently, clustering formulations that aimed to encapsulate *individual* fairness were explored which seek to ensure that each individual is treated fairly. One such formulation was introduced by Jung et al. [17]. This formulation is related to the well-studied k-Center clustering and is the following. Given n points in a metric space representing users, and an integer k, find a set of k centers S such that $d(j, S)$ is at most r_j where r_j denotes the smallest radius around j that contains n/k points. Such a clustering is fair to individual users since no user will be forced to travel outside their *neighborhood*. Jung et al. [17] showed that the problem is NP-Hard and described a simple greedy algorithm that finds k centers S such that $d(j, S) \leq 2r_j$ for all j. Jung et al.'s model can be related to an earlier model of Plesník who considered the Weighted k-Center problem [25]. In Plesník's version, each user j specifies an *arbitrary* radius $r_j > 0$ and the goal is to find k centers S to serve each user within their radius requirement. Plesník showed that a simple variant of a well-known algorithm for k-Center due to Hochbaum and Shmoys [15] yields a 2-approximation. Plesník's problem has been relabeled as the Priority k-Center problem in recent work [6].

Priority clustering. The model of Jung et al. motivated several variations and generalizations of the Priority k-Center problem. Bajpai et al. [6] defined, and provided constant factor approximations, for Priority k-Supplier (where facilities and clients are considered to be disjoint sets), as well as Priority Matroid and Knapsack Center, where facilities are subject to matroid and knapsack constraints, respectively. Mahabadi and Vakilian [23] explored and developed approximation algorithms for Priority k-Median and Priority k-Means problems; their motivation was to combine the individual fairness requirements in terms of radii proposed by Jung et al., with the traditional objectives of clustering. They obtained bicriteria approximation algorithms via local-search. The approximation bounds were later improved via LP-based techniques. Chakrabarty and Negahbani [9] obtained an $(8, 8)$-approximation for Priority k-Median and a $(8, 16)$-approximation for Priority k-Means. Vakilian and Yalcner [28] further improved these results via a nice black box reduction of Priority k-Median to the Matroid Median problem! Via their reduction they obtained $(3, 7.081 + \epsilon)$-approximation for the Priority k-Median problem (relying on the algorithm for Matroid Median from [22]). They extended the algorithmic ideas from Matroid Median to handle ℓ_p norm objectives and were thus able to derive algorithms for Priority k-Means as well. The advantage of the reduction to Matroid Median is the guarantee of 3 on the radius dilation. This is optimal even for the k-Supplier problem [15].

1.2 Results and Technical Contribution

In this paper, we define the PMatMed problem and derive the first (c_1, c_2)-bicriteria approximation algorithms where c_1, c_2 are both constants. There are different trade-offs between c_1 and c_2 that we can achieve. Since PMatMed simultaneously generalizes k-Supplier and Matroid Median, the best c_1 we can hope for is 3, and the best c_2 that we can hope for is ≈ 8, which comes from current algorithms for Matroid Median [22, 27]. We prove the following theorem which captures two results, one optimizing for the radius guarantee, and the other for the cost guarantee.

▶ **Theorem 1.** *There is a $(21, 12)$-approximation algorithm for the Priority Matroid Median Problem. There is also a $(36, 8)$-approximation algorithm.*

As we previously mentioned, [28], via their black box reduction to Matroid Median achieve a $(3, \alpha)$ approximation for Priority k-Median where α is the best approximation for Matroid Median. We conjecture that there is a $(3, O(1))$-approximation for PMatMed. This is interesting and open even for the special case with uniform radii under partition matroid constraint.

Our second set of results are for UniPMatMed. Recall that [18] obtained a $(11, 16)$-approximation for this problem. We prove the following theorem that strictly dominates the bound from [18]. In addition, we show that a tighter radius guarantee is achievable.

▶ **Theorem 2.** *There is a $(9, 8)$-approximation algorithm for the Uniform Priority Matroid Median Problem. For any fixed $\epsilon > 0$ there is a $(5 + 8\epsilon, 4 + \frac{2}{\epsilon})$-approximation.*

▶ Remark 3. We believe that we can extend the ideas from this paper to obtain bicriteria approximation algorithms for Priority Matroid objectives that involve the ℓ_p norm of distances (Priority Matroid Median is when $\ell_p := 1$). Such an approximation algorithm would result in a radius factor dependent on p. [28] already showed that Matroid Median can be extended to the p-norm objective.

Now, we give a brief overview of our technical approach. The reader may wonder about the reduction of Priority k-Median to Matroid Median [28]. Can we make use of this for PMatMed? Indeed one can employ the same reduction, however, the resulting instance is no longer an instance of Matroid Median but an instance of Matroid Intersection Median which is inapproximable [27]. The reduction works in the special case of Priority k-Median since the intersection of a matroid with a cardinality constraint yields another matroid. We therefore address PMatMed directly. Our approximation algorithms are based on a natural LP relaxation. It is not surprising that we need to build upon the techniques for Matroid Median since it is a special case. We build extensively on the LP-based 8-approximation for Matroid Median given by Swamy [27] which improved the first constant factor approximation algorithm of Krishnaswamy et al. [21]. Although the Matroid Median approximation has been improved to 7.081 [22], the approach in [22] seems more difficult to adapt to PMatMed.

Our main technical contribution is to handle the non-uniform radii constraints imposed in PMatMed in the overall approach for Matroid Median. We note that the rounding algorithms for Matroid Median are quite complex, and involve several non-trivial stages: filtering, finding half integral solutions via an auxiliary polytope, and finally rounding to an integral solution via matroid intersection [21, 27, 22]. Kamiyama adapted the ideas in [21] to UniPMatMed and his work involves four stages of reassigments that are difficult to follow. The non-uniform radii case introduces additional complexity. We explain the differences between the uniform radii case and the non-uniform radii case briefly. The LP relaxation opens fractional facilities and assigns each client j to fractionally open facilities. In the LP for PMatMed we write a natural constraint that j cannot be assigned to any facility i where $d(i, j) > r_j$. Let $\bar{C}_j$ denote the distance paid by j in the LP solution. The preceding constraint ensures that $\bar{C}_j \leq r_j$. For UniPMatMed, $r_j = L$ for all $j \in \mathcal{C}$. LP-based approximation algorithms for k-Median use filtering and other rounding steps by sorting clients in increasing order of $\bar{C}_j$ values since they are directly relevant to the objective. When one considers uniform radius constraint, one can still effectively work with $\bar{C}_j$ values since we have $\bar{C}_j \leq L$ for all j. However, when clients have non-uniform radii we can have the following situation; there can be clients j and k such that $\bar{C}_j \ll \bar{C}_k$ but $r_j \gg r_k$. Thus the radius requirements may not correspond to the fractional distances paid in the LP.

We handle the above mentioned complexity via two careful adaptations to Matroid Median rounding. One of these changes occurs in the second stage of Matroid Median rounding, where we construct a half-integral solution using an auxiliary polytope. We must take care to ensure that the half-integral solution constructed in this stage is one that will not violate the radius requirements for clients. To do so, we create additional constraints in the auxiliary polytope. These constraints ensure the half-integral solution satisfies certain properties that are crucial to obtain a constant factor radius guarantee.

The second change occurs in the first filtering stage and plays a role not only for adapting Matroid Median to PMatMed, but also for each of our other results. We first provide an abstract way to describe the filtering stage that allows us to specificy the order in which points are considered, and the distances each point can travel to be reassigned. For our first PMatMed result, the ordering and distances are based on both $\bar{C}_j$ and r_j. For UniPMatMed, we slightly alter the ordering and distances (using the above observations and some ideas from [18]). Our remaining results will also involve changes to the filtering stage. This seems to indicate that filtering plays a large role in the cost and radius trade-off.

Organization. In Section 2, we discuss preliminaries. In particular, we provide definitions and relevant information regarding matroids, define PMatMed and provide its LP relaxation, and discuss the generalized filtering procedure we will use in our algorithm. In Section 3 we present our algorithm for PMatMed and show that it can be used to obtain $(21, 12)$-approximate solutions for instances of PMatMed. In Section 4, we describe how to modify our algorithm for PMatMed to obtain a $(9, 8)$-approximate solution for instances of UniPMatMed, and the remaining results. We also provide some details for the remaining results, and $(36, 8)$-approximate solutions for instances of PMatMed. We defer the proofs of our results to the Appendix.

2 Preliminaries

2.1 Matroids, Matroid Intersection and Polyhedral Results

We assume some basic knowledge about matroids, but provide a few relevant definitions for sake of completeness; we refer the reader to [26] for more details. A matroid $\mathcal{M} = (S, \mathcal{I})$ consists of a finite ground set S and a collection of *independent* sets $\mathcal{I} \subseteq 2^S$ that satisfy the following axioms: (i) $\emptyset \in \mathcal{I}$ (non-emptiness of $\mathcal{I}$) (ii) $A \in \mathcal{I}$ and $B \subset A$ implies $B \in \mathcal{I}$ (downward closure) and (iii) $A, B \in \mathcal{I}$ with $|A| < |B|$ implies there is $i \in B \setminus A$ such that $A \cup \{i\} \in \mathcal{I}$ (exchange property). The rank function of a matroid, $r_{\mathcal{M}} : 2^S \to \mathbb{Z}^{\geq 0}$ assigns to each $X \subseteq S$ the cardinality of a maximum independent subset in X. It is known that $r_{\mathcal{M}}$ is a monotone submodular function. The matroid polytope for a matroid $\mathcal{M}$, denoted by $\mathcal{P}_{\mathcal{M}}$ is the convex hull of the characteristic vectors of the independent sets of $\mathcal{M}$. This can be characterized via the rank function:

$$\mathcal{P}_{\mathcal{M}} = \{v \in \mathbb{R}^S \mid \forall X \subseteq S : \; v(X) \leq r_M(X) \text{ and } \forall e \in S : \; v(e) \geq 0\}.$$

Assuming an independence oracle[1] or a rank function oracle for $\mathcal{M}$, one can optimize and separate over $\mathcal{P}_{\mathcal{M}}$ in polynomial time. A *partition matroid* $\mathcal{M} = (S, \mathcal{I})$ is a special type of matroid that is defined via a partition $S_1, S_2, \ldots, S_h$ of S and non-negative integers $k_1, \ldots, k_h$. A set $X \subseteq S$ is independent, that is $X \in \mathcal{I}$, iff $|X \cap S_i| \leq k_i$ for $1 \leq i \leq h$. A simple partition matroid is one in which $k_i = 1$ for each i.

Given two matroids $\mathcal{M} = (S, \mathcal{I}_1)$ and $\mathcal{N} = (S, \mathcal{I}_2)$, on the same ground set, their intersection is defined as $\mathcal{M} \cap \mathcal{N} := (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ consisting of sets that are independent in both $\mathcal{M}$ and $\mathcal{N}$. Computing a maximum weight independent set in the intersection can be done efficiently. The convex hull of the characteristic vectors of the independent sets of $\mathcal{M} \cap \mathcal{N}$, denoted by $\mathcal{P}_{\mathcal{M},\mathcal{N}}$, is simply the intersection of $\mathcal{P}_{\mathcal{M}}$ and $\mathcal{P}_{\mathcal{N}}$! That is

$$\mathcal{P}_{M,N} = \{v \in \mathbb{R}^S_+ \mid \forall X \subseteq S : \; v(X) \leq r_M(X), v(X) \leq r_{\mathcal{N}}(X)\}.$$

Thus, one can optimize over $\mathcal{P}_{\mathcal{M},\mathcal{N}}$ if one has independence or rank oracles for $\mathcal{M}$ and $\mathcal{N}$. We will need these results later in the paper. See [26] for these classical results.

The input matroid $\mathcal{M}$ for Priority Matroid Median has ground set $\mathcal{F}$ i.e. the set of facilities. Thus, an integer point of the polytope $v^* \in \mathcal{P}_{\mathcal{M}}$ will represent a subset of facilities that is an independent set of the matroid $\mathcal{M}$.

[1] An independence oracle returns whether $A \in \mathcal{I}$ for a given $A \subseteq S$.

2.2 Priority Matroid Median

We provide below a more general definition of Priority Matroid Median that includes a notion of client *demands*.

▶ **Definition 4** (PMatMed). *The input is a set of facilities $\mathcal{F}$ and clients $\mathcal{C}$ from a metric space $(\mathcal{F} \cup \mathcal{C}, d)$. Each $i \in \mathcal{F}$ has an opening cost $f_i \geq 0$. Each client $j \in \mathcal{C}$ has a radius value, $r_j \geq 0$ and a demand value $a_j \geq 0$. We are also given a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$. The goal is to choose a set $S \in \mathcal{I}$ to minimize $\sum_{i \in S} f_i + \sum_{j \in \mathcal{C}} a_j d(j, S))$ with the constraint that $d(j, S) \leq r_j$ for each $j \in \mathcal{C}$.*

A PMatMed instance $\mathscr{I}$ is the tuple $(\mathcal{F}, \mathcal{C}, d, \mathbf{f}, \mathbf{r}, \mathbf{a}, \mathcal{M})$, where $\mathbf{f} \in \mathbb{R}^{\mathcal{F}}$ and $\mathbf{r}, \mathbf{a} \in \mathbb{R}^{\mathcal{C}}$.

2.3 LP relaxation for PMatMed

Our algorithm is based on an LP relaxation for a PMatMed instance $\mathscr{I} = (\mathcal{F}, \mathcal{C}, d, \mathbf{f}, \mathbf{r}, \mathbf{a}, \mathcal{M})$ that we describe next. We use i to index facilities in $\mathcal{F}$, j to index clients in $\mathcal{C}$. Recall that $r_{\mathcal{M}}$ denotes the rank function of the matroid $\mathcal{M}$. The y_i variables denote the fractional amount a facility i is open, while the x_{ij} variables indicate the fractional amount a client j is assigned to facility i.

$$\min \quad \sum_{i \in \mathcal{F}} f_i y_i + \sum_j \sum_i a_j d(i,j) x_{ij} \tag{1a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{F}} x_{ij} \geq 1 \qquad \forall j \in \mathcal{C} \tag{1b}$$

$$x_{ij} \leq y_i \qquad \forall i \in \mathcal{F}, j \in \mathcal{C} \tag{1c}$$

$$x_{ij} = 0 \qquad \forall i \in \mathcal{F}, j \in \mathcal{C} : d(i,j) > r_j \tag{1d}$$

$$y \in \mathcal{P}_{\mathcal{M}} \tag{1e}$$

$$x_{ij}, y_i \geq 0 \qquad \forall i \in \mathcal{F}, j \in C \tag{1f}$$

Constraint 1b states that each client must be fully assigned to facilities, and constraint 1c ensures that these facilities have indeed been opened enough to service clients. For integral y, constraint 1e mandates that the facilities come from an independent set of the matroid $\mathcal{M}$. Finally, constraint 1d ensures that no client is assigned to a center that is farther than its radius value.

We make a few basic observations about the LP relaxation. We assume that it is feasible for otherwise the algorithm can terminate reporting that there is no feasible integral solution. Indeed, the LP is solvable in polynomial time via the rank oracle for $\mathcal{M}$. First, some notation. For $X \subseteq \mathcal{F}$, we let $y(X)$ denote $\sum_{i \in X} y_i$. For client j and radius parameter R we let $B(j, R)$ denote the set $\{i \in \mathcal{F} \mid d(i,j) \leq R\}$ of facilities within R of j. Constraints 1b and 1d ensure the following simple fact.

▶ **Fact 5.** *Let (x, y) be a feasible solution to the PMatMed LP. Then $y(B(j, r_j)) \geq 1$ holds $\forall j \in \mathcal{C}$.*

Let $COST(x, y)$ denote the cost of the LP using solution (x, y). Going forward, we will assume that we are working with an optimum fractional solution to the LP relaxation for the given instance.

▶ Remark 6. We say that y is feasible if $y \in \mathcal{P}_{\mathcal{M}}$ and $y(B(j, r_j)) \geq 1$ for all $j \in C$. Given feasible y, a corresponding x satisfying the constraints and minimizing $COST(x, y)$ is determined by solving a min-cost assignment problem for each client $j \in C$ separately.

2.4 Filtering

Filtering is a standard step in several approximation algorithms for clustering and facility location wherein one identifies a subset of well-separated and representative clients. Each client is assigned to a chosen representative. In priority median problems there are two criteria that dictate the filtering process. One is the radius upper bound r_j for the client j. The other is the LP distance $\bar{C}_j = \sum_i d(i,j)x(i,j)$ paid by the client which is part of the objective. Balancing these two criteria is important. To facilitate different scenarios later we develop a slightly abstract filtering process. Building on a procedure introduced in [15, 25], Filter takes in the metric and demands from a PMatMed instance $\mathscr{I} = (\mathcal{F}, \mathcal{C}, d, \mathbf{f}, \mathbf{r}, \mathbf{a}, \mathcal{M})$, as well as functions $\phi, \lambda : \mathcal{C} \to \mathbb{R}_+$ that satisfy the following condition.

▶ **Definition 7** (compatibility). *Functions $\phi, \lambda : \mathcal{C} \to \mathbb{R}_+$ are compatible if for any ordering of clients $j_1, j_2, \ldots, j_n$ where $\phi(j_1) \le \phi(j_2) \le \ldots \le \phi(j_n)$, it is the case that $\lambda(j_1) \le \lambda(j_2) \le \ldots \le \lambda(j_n)$.*

▶ Remark 8. This condition trivially holds when ϕ and λ are identical. The filtering stages of many clustering approximation algorithms [6, 17, 9] utilize equal ϕ and λ functions. We use both identical and non-identical settings for ϕ and λ in this paper.

The function ϕ encodes an ordering of clients, while λ represents a client's coverage distance. Filter chooses cluster centers in order of increasing ϕ values, and then "covers" any remaining client k that is within distance $2 \cdot \lambda(k)$ from the newly added center j. The demand from the covered points is transferred to the center that first covered them. The new demand variables a' represent the aggregated demand for the chosen centers. Filter returns the set of cluster centers, the clusters assigned to each cluster center, and new demand assignments for all clients.

Algorithm 1 Filter.

Require: Metric $(\mathcal{F} \cup \mathcal{C}, d)$, demands $\mathbf{a}$, compatible functions $\phi, \lambda : \mathcal{C} \to \mathbb{R}_{>0}$

$U \leftarrow \mathcal{C}$ ▷ The set of uncovered clients
$C \leftarrow \emptyset$ ▷ The set of cluster centers
$\forall j \in \mathcal{C}$ set $a'_j := 0$ ▷ Initialize new demand variables
while $U \neq \emptyset$ **do**
    $j \leftarrow \arg\min_{j \in U} \phi(j)$
    $C \leftarrow C \cup \{j\}$
    $D(j) \leftarrow \{k \in U : d(j,k) \le 2 \cdot \lambda(k)\}$ ▷ Note: $D(j)$ includes j itself
    $a'_j = \sum_{k \in D(j)} a_k$ ▷ Accumulate all demands of $D(j)$ to j
    $U \leftarrow U \backslash D(j)$
end while
Return cluster centers C, $\{D(j) : j \in C\}$, updated demands $\mathbf{a}' \in \mathbb{R}^{\mathcal{C}}$

The resulting cluster centers $C \subseteq \mathcal{C}$, and the sets of clients relocated to each cluster center $\{D(j) \mid j \in C\}$ form a partition of the client set $\mathcal{C}$. When the given ϕ and λ are compatible, the returned clusters satisfy certain desirable properties, described in the following facts which are relatively easy to see, and standard in the literature. For this reason we omit formal proofs.

▶ **Fact 9.** *The following statements hold for the output of Filter: (a) $\forall j, j' \in C, d(j,j') > 2\max\{\lambda(j), \lambda(j')\}$. (b) $\{B(j, \lambda(j)) \mid j \in C\}$ are mutually disjoint. (c) $\{D(j) \mid j \in C\}$ partitions $\mathcal{C}$. (d) $\forall j \in C, \forall k \in D(j), \phi(j) \le \phi(k)$ and $\lambda(j) \le \lambda(k)$. (e) $\forall j \in C, \forall k \in D(j), d(j,k) \le 2 \cdot \lambda(k)$*

Choosing ϕ and λ. As we remarked, the two criteria that influence the filtering process are r_j and $\bar{C}_j$. For the algorithm in Section 3 we choose $\phi(j) = \lambda(j) = \min\{r_j, 2\bar{C}_j\}$. There are other valid settings of compatible ϕ and λ that can be used in the filtering stage. Different settings of ϕ and λ will result in different approximation factors for cost and radius. We elaborate on this further in Section 4.

3 A $(21, 12)$-approximation for Priority Matroid Median

Our algorithm will follow the overall structure of the LP-based procedure used for approximating Matroid Median from [27], but will contain a few key alterations that allow us to be mindful of the radius objective of PMatMed. Stage 1 of our algorithm involves filtering the client set to construct an updated instance $\mathscr{I}'$ using the cluster centers and updated demands. We will show that a solution to $\mathscr{I}'$ can be converted to a solution for $\mathscr{I}$ while only incurring a small increase to the cost and radius. The focus then shifts to constructing a solution for $\mathscr{I}'$. In Stage 2, we obtain a half-integral solution for the LP-relaxation for $\mathscr{I}'$ by working with an auxiliary polytope. In Stage 3, this half-integral solution is converted to an integral solution for $\mathscr{I}'$. This is done via a reduction to matroid intersection. Finally, we will show that this solution yields a $(21, 12)$-approximation for the original instance $\mathscr{I}$. Algorithm 2 is given as a summary of the various stages of our algorithm. The omitted proofs from this section can be found in Appendix A.

Algorithm 2 Overview of bi-criteria approximation algorithm for PMatMed.

Input: PMatMed instance $\mathscr{I} = (\mathcal{F}, \mathcal{C}, d, \mathbf{f}, \mathbf{r}, \mathbf{a}, \mathcal{M})$.
Output: (α, β)-approximate solution for $\mathscr{I}$.

0: Solve LP for $\mathscr{I}$ and let (x, y) denote the optimal fractional solution. Use (x, y) and radius values $\mathbf{r}$ to help set ϕ and λ.
1: **Stage 1 -** Run Filter$((\mathcal{F} \cup \mathcal{C}, d), \mathbf{a}, \phi, \lambda)$ which returns cluster centers C, and updated client demands $\mathbf{a}'$. Create an updated instance $\mathscr{I}' = (\mathcal{F}, C, d, \mathbf{f}, \mathbf{r}, \mathbf{a}', \mathcal{M})$ (Section 3.1).
2: **Stage 2 -** Construct a half-integral solution $(\hat{x}, \hat{y})$ for $\mathscr{I}'$ by setting up a polytope $\mathcal{Q}$ with half-integral extreme points (Section 3.2).
3: **Stage 3 -** Convert the half-integral solution to an integral solution $(\tilde{x}, \tilde{y})$ for $\mathscr{I}'$ by setting up an instance of matroid intersection between the input matroid $\mathcal{M}$, and a partition matroid $\mathcal{N}$ constructed with respect to the half-integral solution (Section 3.3).
4: Convert the integral solution for $\mathscr{I}'$ to one for $\mathscr{I}$ (Lemma 10).

3.1 Stage 1: Filtering Clients

In this stage, we create a new instance of PMatMed from the initial one by using the Filter process described in Section 2.4. Recall that Filter will return a set of cluster centers $C \subseteq \mathcal{C}$, and collections of clients that are relocated to each cluster center $\{D(j) \mid j \in C\}$. Filter also returns a set of updated demands for all clients, $\mathbf{a}'$. Now, using C and $\mathbf{a}'$, we construct a new instance of PMatMed $\mathscr{I}' = (\mathcal{F}, C, d, \mathbf{f}, \mathbf{r}, \mathbf{a}', \mathcal{M})$. Here, we overload notation and take $\mathbf{r}$ and $\mathbf{a}'$ to denote the vector of radius values and demands, respectively, for cluster centers (i.e. $\mathbf{r}, \mathbf{a}' \in \mathbb{R}^C$). Notice that we do not lose any information by restricting $\mathbf{a}'$ to C, since the updated demands for relocated points are set to 0. Furthermore, we will reconcile the radius objective for relocated points in the final solution at the end of the section.

The solution (x, y) for instance $\mathscr{I}$, when restricted to C, will still be a feasible solution for the LP for $\mathscr{I}'$, since the new LP is made up of a subset of constraints from the original LP. For updated instance $\mathscr{I}'$, we denote the cost of the LP solution (x, y) by $COST'(x, y)$.

$$COST'(x,y) = \sum_{i \in \mathcal{F}} f_i y_i + \sum_{j \in C} a'_j \sum_{i \in \mathcal{F}} d(i,j) x_{ij} = \sum_{i \in \mathcal{F}} f_i y_i + \sum_{j \in C} a'_j \bar{C}_j$$

The next lemma shows that an integral solution to $\mathscr{I}'$ can be translated to an integer solution for $\mathscr{I}$ by incurring a small additive increase to the cost objective. In subsequent sections we will address how the translated solution also ensures that all clients are served within a constant factor of their radius constraint.

▶ **Lemma 10.** *The following is true of $\mathscr{I}'$: (a) $COST'(x, y) \leq 2 \cdot COST(x, y)$. (b) Any integer solution (x', y') for $\mathscr{I}'$ can be converted to an integer solution for $\mathscr{I}$ that incurs an additional cost of at most $4 \cdot COST(x, y)$.*

The following lemma follows directly from Fact 9.

▶ **Lemma 11.** *Let $k \in \mathcal{C}$ be assigned to $j \in C$ after* Filter *(i.e. $k \in D(j)$). Then, $d(j, k) \leq 2\lambda(k) \leq 2r_k$.*

3.2 Stage 2: Constructing Half-Integral Solution $(\hat{x}, \hat{y})$

In the second stage the goal is to construct a half-integral solution to $\mathscr{I}'$. This means that each cluster center/client $j \in C$ will connect to at most two facilities. This is accomplished by constructing a specific polytope $\mathcal{Q}$ with only facility variables, and a proxy objective that also has only facility variables and arguing about the properties of $\mathcal{Q}$ and the objective function.

To describe $\mathcal{Q}$, we define, for each client $j \in C$, several facility sets that will play an important role. Let $F_j = \{i \in \mathcal{F} \mid d(i, j) = \min_{k \in C} d(i, k)\}$ denote the set of facilities i for which j is the closest client in C (ties are broken arbitrarily). Let $F'_j = \{i \in F_j \mid d(i, j) \leq \lambda(j)\} \subseteq F_j$. Let $\gamma_j := \min_{i \notin F_j} d(i, j)$ denote the distance between client $j \in C$ and the closest facility i not included in F_j. In other words, i in the definition of γ_j is the closest facility to j that has some other closest cluster center $j' \in C$ such that $j \neq j'$. Using γ_j, let $G_j = \{i \in F_j \mid d(i, j) \leq \gamma_j\}$. Finally, let ρ_j be the smallest distance such that $y(B(j, \rho_j)) \geq 1$, and $B_j := B(j, \rho_j)$.[2] See Figure 1.

We summarize some basic properties of the defined sets below.

▶ **Fact 12.** *The following hold for all $j \in C$: (a) If $j' \neq j$, $F_j \cap F_{j'} = \emptyset$; (b) F_j contains all the facilities i such that $d(i, j) \leq \lambda(j)$; (c) $\gamma_j > \lambda(j)$; (d) $F'_j \subseteq G_j$; (e) $\rho_j \leq r_j$, (f) $\sum_{i \in F'_j} x_{ij} \geq 1/2$ and when $\lambda(j) = r_j$, $\sum_{i \in F'_j} x_{ij} = 1$;*

Proof. (a) follows from definition of F_j, (b), (c), (d) follow from Fact 9(b) and definitions. (e) follows from the LP constraint. We now prove (f). If $\lambda(j) = r_j$, $F'_j = \{i \mid d(i, j) \leq r_j\}$, and by LP constraint $\sum_{i \in F'_j} x_{ij} = 1$. Otherwise $\lambda(j) = 2\bar{C}_j < r_j$. Note that $\bar{C}_j = \sum_i d(i, j) x_{ij}$. By averaging argument (Markov's inequality) we have $\sum_{i: d(i,j) \leq 2\bar{C}_j} x_{ij} \geq 1/2$. This gives the desired claim since $F'_j = \{i \mid d(i, j) \leq \lambda(j)\}$. ◀

[2] Note that though it may be the case that $y(B(j, \rho_j)) > 1$, we can split facilities and define B_j as the points of $B(j, \rho_j)$ such that $y(B_j) = 1$.

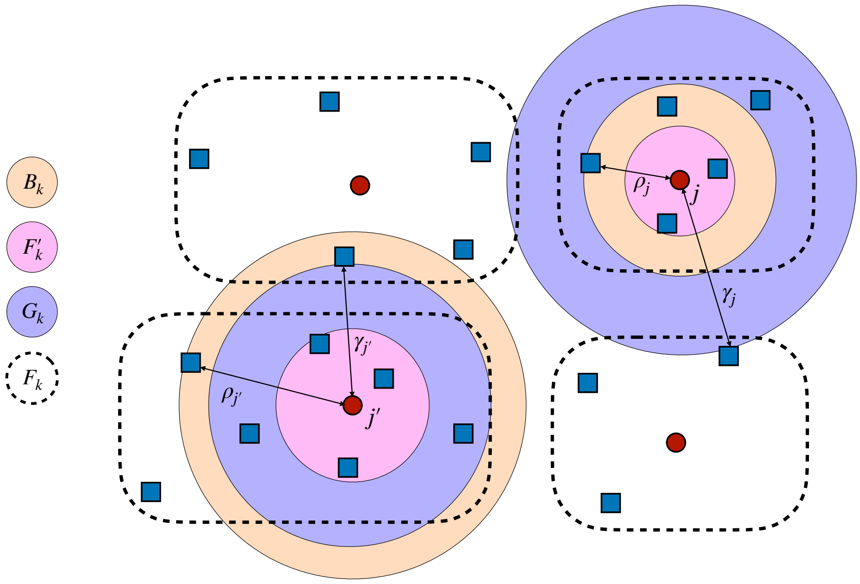

Figure 1 The F, F', G, and B sets for points $j \in C_s$ and $j' \in C_b$. Observe that for j, $\rho_j \leq \gamma_j$, hence $B_j \subseteq G_j$.

At this point in the algorithm, in a departure from the Matroid Median algorithm of [27], we need to be mindful of two cases. If $\rho_j \leq \gamma_j$, in order to satisfy the radius requirements of PMatMed, it is important to open one facility within radius ρ_j of j. If it is the case that $\rho_j > \gamma_j$, it is not necessary to do so. To distinguish these two cases, we partition C into $C_s = \{j \in C \mid \rho_j \leq \gamma_j\}$, and $C_b = \{j \in C \mid \rho_j > \gamma_j\}$. For $j \in C_s$, it should be clear that $B_j \subseteq G_j$. Using these sets, we define a polytope $\mathcal{Q}$ with facility variables $v_i, i \in \mathcal{F}$ as follows. It consists of the matroid constraints induced by $\mathcal{M}$ and a second set of constraints induced by C and C_s as defined above. In particular, we require that all points j in C has at least $1/2$ value assigned cumulatively to facilities within their F'_j balls. We require points of C_s to have exactly 1 assigned to facilities within B_j.

$$\mathcal{Q} = \Big\{ v \in \mathbb{R}^{\mathcal{F}}_{\geq 0} \ \mid \ \forall S \subseteq \mathcal{F} : v(S) \leq r_{\mathcal{M}}(S), \ \ \forall j \in C : \ v(F'_j) \geq 1/2 \text{ and } v(G_j) \leq 1, \\ \forall j \in C_s : v(B_j) = 1 \Big\}$$

▶ **Lemma 13.** *The extreme points of the polytope $\mathcal{Q}$, if non-empty, are half-integral.*

The proof of the preceding lemma is similar to those in previous works on Matroid Median [21, 27]. We give a proof (found in Appendix A) for the sake of completeness since the polytope we define is slightly different due to the separation of clients in C into C_s and C_b in order to enforce an additional constraint.

We will now define a vector y' that lies in $\mathcal{Q}$ which will prove that it is non-empty. Further, we also define a linear objective function $T(\cdot)$ over vectors in $\mathcal{Q}$ to serve as a proxy for the cost. Following the analysis for the improved bound in [27], we set up $T(\cdot)$ with some slack so that the slack can be exploited in the analysis of the next step in the algorithm.

We define $y' \in \mathbb{R}^{\mathcal{F}}_{\geq 0}$ as follows. For all $j \in C$ and $i \in G_j$, set $y'_i = x_{ij} \leq y_i$. For a facility $i \notin \cup_j G_j$ set $y'_i = 0$. From this definition it should be clear that $y'(G_j) \leq 1$ for all $j \in C$, since $\sum_{i \in G_j} x_{ij} \leq 1$. Also, from Fact 12, $y'(F'_j) \geq 1/2$. For $j \in C_s$, it will be the case that $y'(B_j) = 1$ since $\sum_{i \in B_j} x_{ij} = 1$; we also know that for these points, $y'(G_j) = y'(B_j)$.

To build up to the definition of T, we first state the following lemma, which we will prove in the proof of Lemma 17 (Appendix A).

▶ **Lemma 14.** *Consider some $j \in C$, and let i and j' be the facility and cluster used to define γ_j (i.e. $\gamma_j = d(i,j)$) where $i \in F_{j'}$ for some $j' \neq j$. For every $i' \in F'_{j'}$, $d(i',j) \leq 3\gamma_j$.*

Keeping the preceding lemma in mind, we can use as proxy for j's per-unit-demand cost a function written in terms of the facility vector v. When $y'(G_j) = 1$, the cost for j can be bounded by $\sum_{i \in G_j} d(i,j)y'_i \leq \bar{C}_j$. When $y'(G_j) < 1$, the preceding lemma indicates that we can upper bound the cost of the solution by $\sum_{i \in G_j} d(i,j)y'_i + 3\gamma_j(1 - y'(G_j)) \leq 3 \cdot \bar{C}_j$. Using these two bounds, we define $T(\cdot)$ for $v \in \mathcal{Q}$ as follows:

$$T(v) = \sum_{i \in \mathcal{F}} f_i v_i + \sum_{j \in C} a'_j \Big(2 \sum_{i \in G_j} d(i,j) v_i + 4\gamma_j (1 - v(G_j)) \Big)$$

For v such that $v(F'_j) \geq 0.5$ and $v(G_j) \leq 1$ for all $j \in C$, the term $a'_j(2\sum_{i \in G_j} d(i,j)y'_i + 4\gamma_j(1 - y'(G_j)))$ will upper bound j's assignment cost with respect to v via Lemma 14. When $v(G_j) = v(B_j) = 1$ for $j \in C_s$, j's assignment cost will be at most $a'_j(2\sum_{i \in B_j} d(i,j)v_i)$. Indeed $T(v)$ is an *overestimate* and we will use this in the next step.

We find an optimum half-integral solution $\hat{y}$ to $\mathcal{Q}$ with objective $T(v)$. It follows that $T(\hat{y}) \leq T(y')$. Now, we construct a half-integral solution $(\hat{x}, \hat{y})$ from $\hat{y} \in \mathcal{Q}$: For each cluster center $j \in C$, if $\hat{y}(G_j) = 1$, set $\sigma(j) = j$. Otherwise, set $\sigma(j) = \arg\min_{j' \in C : j' \neq j} d(j,j')$. Now, the *primary facility* for each cluster center is the closest facility $i \in \mathcal{F}$ such that $\hat{y}_i > 0$ (this will always be located in F'_j), is denoted by $i_1(j)$, and thus $\hat{x}_{i_1(j)j} = \hat{y}_{i_1(j)}$. A cluster center's *secondary* facility, denoted by $i_2(j)$, is the next option of facility for j to use, when it cannot be completely serviced by its primary facility. If $\hat{y}_{i_1(j)} = 1$, then j does not need a secondary facility, since $i_1(j)$ has been completely opened, and will remain completely opened. When $\hat{y}_{i_1(j)} < 1$ and $\hat{y}(G_j) = 1$, then set $i_2(j)$ to be the second closest partially opened facility to j (where $\hat{y}_{i_2(j)} > 0$). Otherwise, when $\hat{y}_{i_1(j)} < 1$ and $\hat{y}(G_j) < 1$, we now set $i_2(j) = i_1(\sigma(j))$ and $\hat{x}_{i_1(j)} = \hat{x}_{i_2(j)} = 1/2$. Note that if $j \in C_s$ then $\hat{y}(B_j) = 1$ which implies that j's primary and secondary facilities are both in B_j and $\sigma(j) = j$. The following two claims are easy to see.

▷ Claim 15. For all $j \in C$, $d(j, \sigma(j)) \leq 2\gamma_j$.

▷ Claim 16. For all $j \in C_s$, $\hat{y}(G_j) = 1$. If $\hat{y}(G_j) < 1$, it must be the case that $j \in C_b$.

By Fact 9(b), each j will have a unique primary facility that is at least partially opened in F'_j. For points $j \in C_s$, their secondary facility must be in B_j. However, for points in $j \in C_b$, $i_2(j)$ might not be in G_j or even F_j. As per Lemma 14, we know that j will be able to find a partially open facility to be serviced by that is within distance $3\gamma_j < 3\rho_j$. In the following lemma, we derive our bound for the cost of $(\hat{x}, \hat{y})$.

▶ **Lemma 17.** *$COST'(\hat{x}, \hat{y}) \leq T(\hat{y}) \leq T(y') \leq 4 \cdot COST'(x,y) \leq 8 \cdot COST(x,y)$.*

Before moving on to the final stage of the algorithm, we prove a few lemmas that will be relevant for our analysis of the radius dilation of the final solution. Lemma 18 allows us to relate the radius of cluster center j to that of a client k in the original instance that was relocated to j. We need such a lemma because even though we know that $\phi(j) = \min\{r_j, 2\bar{C}_j\}$ and $\phi(j) \leq \phi(k)$ for all $k \in D(j)$, we cannot assume that $r_j \leq r_k$.

▶ **Lemma 18.** *Suppose client $k \in \mathcal{C}$ is relocated to $j \in C$ after filtering ($k \in D(j)$). Then $\rho_j \leq 3r_k$.*

Proof. Note that $y(B(k, r_k)) \geq 1$ via the LP constraint. We have $d(j,k) \leq 2\lambda(k) \leq 2r_k$ since $\lambda(k) = \min\{r_k, 2\bar{C}_k\}$. Via triangle inequality, $B(k, r_k) \subseteq B(j, 3r_k)$. Thus $\rho_j \leq 3r_k$. ◀

Lemma 14 and Lemma 18 imply Lemma 19, which bounds the distance between relocated points and the primary and secondary facilities of the cluster center they are relocated to.

▶ **Lemma 19.** *Let $k \in \mathcal{C}$ and $k \in D(j)$ for a cluster center $j \in C$. Then, $d(j, i_1(j)) \leq \lambda(j) \leq \lambda(k) \leq r_k$. When $j \in C_s$, $d(j, i_2(j)) \leq \rho_j \leq 3r_k$. When $j \in C_b$, $d(j, i_2(j)) \leq d(j, \sigma(j)) + d(i_1(\sigma(j)), \sigma(j)) \leq 3\gamma_j \leq 3\rho_j \leq 9r_k$.*

▶ Remark 20. Notice that the $v(B_j) = 1$ constraint imposed for points $j \in C_s$ ultimately did not effect the cost analysis in Lemma 17. That is, we did not need to draw a distinction between points in C_s and points in C_b in order to obtain $COST'(\hat{x}, \hat{y}) \leq 4 \cdot COST'(x, y)$. The purpose of defining sets C_s and C_b and imposing an additional constraint for points in C_s is to ensure certain radius guarantees. In particular, Lemma 19 would not hold if the constraint $v(B_j) = 1$ for $j \in C_s$ was not enforced in $\mathcal{Q}$.

3.3 Stage 3: Converting to an Integral Solution

The procedure to convert the half-integral $(\hat{x}, \hat{y})$ to an integral solution involves setting up a matroid intersection instance consisting of the input matroid $\mathcal{M}$ and a partition matroid that is constructed using the primary and secondary facilities from $(\hat{x}, \hat{y})$ after another clustering step. The solution to this instance will be used to construct an integral solution $(\tilde{x}, \tilde{y})$ to $\mathscr{I}'$.

For $j \in C$ set $\hat{C}_j = (d(i_1(j), j) + d(j, \sigma(j)) + d(i_2(j), \sigma(j)))/2$. In cases where j has no secondary facility, let $i_2(j) = i_1(j)$. For each $j \in C$, define $S_j = \{i \mid \hat{x}_{ij} > 0\} = \{i_1(j), i_2(j)\}$. S_j has either one or two facilities. In addition, the following holds and will be relevant later.

▷ Claim 21. When $S_j \cap S_{j'} \neq \emptyset$, one of three cases can occur. (i) $S_j \cap S_{j'} = \{i_1(j), i_2(j)\}$, in which case $\sigma(j) = j'$ and $\sigma(j') = j$; (ii) $S_j \cap S_{j'} = \{i_1(j)\}$, and thus $\sigma(j') = j$ and $\sigma(j) \neq j'$ (a symmetric case occurs when switching j and j'); (iii) $S_j \cap S_{j'} = \{i_2(j)\}$ where $i_2(j) = i_2(j')$, hence $\sigma(j) = \sigma(j') = p$ and $p \neq j, j'$.

We construct a partition matroid $\mathcal{N}$ on ground set $\mathcal{F}$ via another clustering process to create a set $C' \subseteq C$. Repeat the following two steps until no clients in C are left to consider: (1) Pick $j \in C$ with the smallest $\hat{C}_j$ value and add j to the set C' then (2) remove every $j' \in C$ where $S_j \cap S_{j'} \neq \emptyset$, and have j be the center of j' (denoted by $\mathsf{ctr}(j') = j$). It is easy to see that the sets $S_j, j \in C'$ are mutually disjoint. Thus, a partition of $\mathcal{F}$ is induced by $\{S_j \mid j \in C'\}$, and the set $\mathcal{F} \setminus \cup_{j \in C'} S_j$. Set the capacity for each set of this partition to 1.

Now we consider the polytope that is intersection of the matroid polytopes of $\mathcal{M}$ and $\mathcal{N}$:

$$\mathcal{R} = \{z \in \mathbb{R}_+^{\mathcal{F}} \mid \forall S \subseteq \mathcal{F}: \ z(S) \leq r(S), \ \ \forall j \in C': \ z(S_j) \leq 1\}$$

The polytope $\mathcal{R}$ has integral extreme points via the classical result of Edmonds [12, 26].

The goal now is to figure out the set of facilities to open by optimizing a relevant objective over $\mathcal{R}$. First, we define a vector $\hat{y}' \in \mathbb{R}_+^{\mathcal{F}}$: if $i \in S_j$ for some $j \in C'$ we set $\hat{y}'_i = \hat{x}_{ij} \leq \hat{y}_i$, otherwise we set $\hat{y}'_i = \hat{y}_i$. Observe that $\hat{y}'$ is feasible for $\mathcal{R}$ and shows that $\mathcal{R}$ is not empty.

We now define a linear function $H(\cdot)$ over vectors in $\mathcal{R}$. We will optimize $H(\cdot)$ over $\mathcal{R}$ to obtain an integral extreme point $\tilde{y}$ and we will analyze its cost via $\hat{y}'$. For $z \in \mathbb{R}_+^{\mathcal{F}}$, define $H(z)$ as follows.

$$H(z) = \sum_i f_i z_i + \sum_{j \in C} L_j(z), \text{ where}$$

$$L_j(z) = \begin{cases} \sum_{i \in S_{\mathsf{ctr}(j)}} a'_j d(i,j) z_i & i_1(j) \in S_{\mathsf{ctr}(j)} \\ \sum_{i \in S_{\mathsf{ctr}(j)}} a'_j \Big(d(j,\sigma(j)) + d(\sigma(j),i) \Big) z_i & \\ \qquad + a'_j \Big(d(i_1(j),j) - d(j,\sigma(j)) - d(i_1(\sigma(j)),\sigma(j)) \Big) z_{i_1(j)} & \text{otherwise} \end{cases}$$

Let $\tilde{y} \in \mathcal{R}$ be an integer extreme point such that $H(\tilde{y}) \leq H(\hat{y}')$. We use this to define an integral solution $(\tilde{x}, \tilde{y})$ to the modified instance by assigning each $j \in C'$ to the facility opened from S_j i.e. the facility $i \in S_j$ such that $\tilde{y}_i = 1$. For each $j' \in C \setminus C'$, assign j' to either $i_1(j')$ if it is open or the facility opened from $S_{\mathsf{ctr}(j')}$. $L_j(\tilde{y})$ serves as a proxy and upper bound for j's assignment cost. When $i_1(j) \notin S_{\mathsf{ctr}(j)}$, the second term of $L_j(\tilde{y})$ will adjust the distance j pays depending on whether $i_1(j)$ is opened or not. This adjustment is not needed when $i_1(j) \in S_j$ or when $i_1(j) \notin S_j$ is not opened, since in this case j must be assigned to the center opened from $S_{\mathsf{ctr}(j)}$. The following lemmas will show how the cost of $(\tilde{x}, \tilde{y})$ can be bounded by that of the half-integral solution $(\hat{x}, \hat{y})$ from the previous stage.

▶ **Lemma 22.** *$COST'(\tilde{x}, \tilde{y})$ is at most $H(\tilde{y}) \leq H(\hat{y}')$.*

▶ **Lemma 23.** *$H(\hat{y}') \leq T(\hat{y})$.*

▶ Remark 24. We do not lose a factor in the cost when converting the half-integral solution to an integral solution because the analysis in Stage 2 "overpays" for the half-integral solution. We follow the approach from [27].

3.4 Cost and Radius Analysis for PMatMed

Lemmas 10, 17, 22, and 23 together imply the following bound on the cost of $(\tilde{x}, \tilde{y})$ for instance $\mathscr{I}$ with respect to the cost of the LP solution (x, y).

▶ **Theorem 25.** *$COST(\tilde{x}, \tilde{y}) \leq 12 \cdot COST(x, y)$.*

Proof. $COST'(\tilde{x}, \tilde{y})$ will be at most $T(\hat{y})$ (Lemmas 22 and 23), and $T(\hat{y})$ is at most $4 \cdot COST'(x, y) \leq 8 \cdot COST(x, y)$ (Lemma 17). Hence, $(\tilde{x}, \tilde{y})$ will give a solution to $\mathscr{I}'$ of cost at most $8 \cdot COST(x, y)$. Lemma 10 tells us that translating an integer solution for $\mathscr{I}'$ to an integer solution for $\mathscr{I}$ will incur an additional cost of at most $4 \cdot COST(x, y)$. All together, $COST(\tilde{x}, \tilde{y}) \leq COST'(\tilde{x}, \tilde{y}) + 4 \cdot COST(x, y) \leq 8 \cdot COST(x, y) + 4 \cdot COST(x, y) = 12 \cdot COST(x, y)$. ◀

To complete our analysis of the radius approximation factor, we must determine how far points will be made to travel once the final centers are chosen. In Lemma 19 we guaranteed that each cluster center j will not travel farther than $3\rho_j$ to reach its secondary facility. However, in this final stage, we are assigning some cluster centers to others, and cannot guarantee that their primary or secondary facility will be opened. We can still show that even if a cluster center j from C_s gets assigned to a cluster center ℓ from C_b (i.e. that $\mathsf{ctr}(j) = \ell$), j will still only travel a constant factor outside of ρ_j. Consequently, using Lemma 18 we can show that each client $k \in \mathcal{C}$ will travel only a constant factor times its radius value r_k.

▶ **Lemma 26.** *Let $k \in \mathcal{C}$, where $k \in D(j)$ for $j \in C$. The final solution will open a facility i such that $d(i, j) \leq 19 r_k$.*

Proof. There are several cases to consider but most of them are simple. We provide the analysis for the case that gives the 19 factor, and other notable cases.

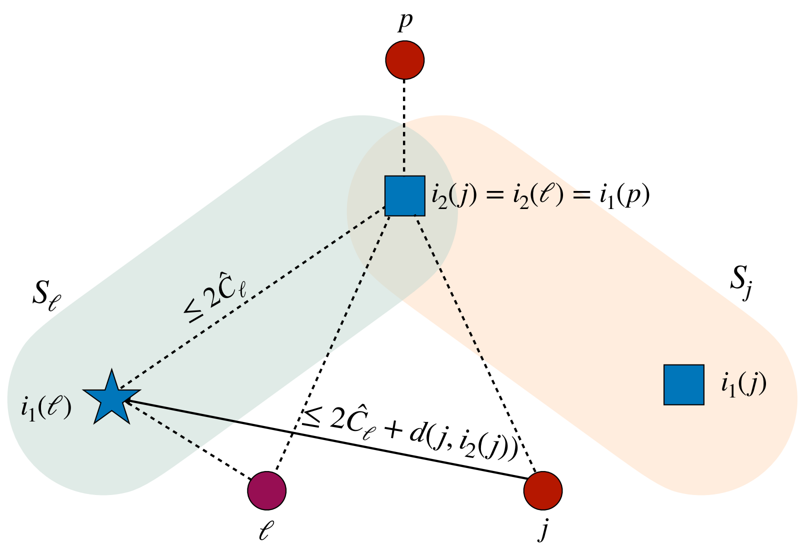

Figure 2 The farthest a point $j \in C$ will be from an opened center occurs when $\mathsf{ctr}(j) = \ell$, $\sigma(j) = \sigma(\ell) = p$, and $i_1(\ell)$ is opened.

If $j \in C'$, then either $i_1(j)$ or $i_2(j)$ will be opened in the final solution. Lemma 19 indicates that j will be assigned to a center that is at most $9r_k$ away. If $j \notin C'$, it must be the case that $\mathsf{ctr}(j) = \ell$ where $S_\ell \cap S_j \neq \emptyset$, and $\hat{C}_\ell \leq \hat{C}_j$. We claim that $\hat{C}_j \leq \frac{1}{2}(d(i_1(j), j) + d(i_2(j), j)) \leq \frac{1}{2}(r_k + 9r_k) = 5r_k$ where we used Lemma 19 to bound $d(i_1(j), j)$ and $d(i_2(j), j)$).

The farthest that j would have to travel occurs when j and ℓ share secondary facilities, and ℓ's primary facility is opened (see Figure 2). More precisely, this is when $S_\ell \cap S_j = \{i_2(\ell)\} = \{i_2(j)\}$ and $\sigma(\ell) = \sigma(j) = p$ where p is not j or ℓ, and $i_1(\ell)$ is opened at the end of Stage 3. In this case, we have

$$\begin{aligned} d(i_1(\ell), j) &\leq d(i_1(\ell), i_2(\ell)) + d(i_2(\ell), j) \leq d(i_1(\ell), \ell) + d(i_2(\ell), \ell) + d(i_2(j), j) \\ &= 2\hat{C}_\ell + d(i_2(j), j) \leq 2\hat{C}_j + d(i_2(j), j) \leq 10r_k + 9r_k = 19r_k. \end{aligned}$$ ◀

▶ Remark 27. Notice that in the last step of our proof for Lemma 26, we bound the distance $d(i_1(\ell), j)$ by $d(i_1, j) + 2d(i_2(j), j)$, where $d(i_2(j), j) \leq 9r_k$. Hence, the majority of the distance that j is traveling, according to our analysis, is due to the distance between j and its secondary facility. If we could guarantee that cluster center j has a reasonably close secondary facility, we could improve this radius factor. We will explore this further in Section 4.3.

Using Lemmas 11 and 26, we have the following radius bound for the output of our algorithm.

▶ **Theorem 28.** *Let S be the output of the aforementioned approximation algorithm. For all $k \in \mathcal{C}$, $d(k, S) \leq 21r_k$.*

Theorems 25 and 28 together give us Theorem 1.

4 Exploring Cost and Radius Trade-offs

In this section, we will outline the remaining results for PMatMed and UniPMatMed. The algorithms for these results are nearly identical to the one in the previous section. The only change is in setting ϕ and λ (in the first step of Algorithm 2). These results suggest that filtering plays a non-trivial role in the cost and radius trade-offs, and that further improvements may be possible if one finds effective ways to filter points.

We begin with the $(9, 8)$-approximate solution for UniPMatMed. As we discussed in the introduction, having uniform radii allows us to rely only on $\bar{C}_j$ values obtained from the LP solution. We will then discuss how we can extend this approach to the non-uniform case to obtain a $(36, 8)$-approximate solution for PMatMed. Finally, we show how to further tighten the radius guarantee for UniPMatMed.

4.1 $(9, 8)$-approximation for UniPMatMed

First, observe that instances of UniPMatMed can be written as instances of PMatMed, where each $r_j := L$ for all $j \in \mathcal{C}$. As such, our algorithm to obtain a $(9, 8)$-approximation for UniPMatMed is the following: Run Algorithm 2 on UniPMatMed instance $\mathscr{J}$, but in Line 0, set $\phi(j) := \bar{C}_j$ and $\lambda(j) := \min\{r_j, 2\bar{C}_j\}$.

Notice that these assignments of ϕ and λ satisfy compatibility (Definition 7) only when r_j's are uniform. We explain this in more detail in Appendix B, and defer the analysis and proofs for this result to that section.

4.2 $(36, 8)$-approximation for PMatMed

Building off the result for UniPMatMed, which is able to optimize for the cost by changing $\phi(j)$, we show how to obtain a $(36, 8)$-approximate solution for PMatMed. To do so, we will keep the same setting for $\phi(j) := \bar{C}_j$ as UniPMatMed, but will instead choose a λ that is compatible for non-uniform radii. Our algorithm is as follows: Run Algorithm 2 on PMatMed instance $\mathscr{I}$, but in Line 0, set $\phi(j) := \bar{C}_j$ and $\lambda(j) := 2\bar{C}_j$.

Clearly, ϕ and λ are compatible. Furthermore, notice that this setting of ϕ is identical to that of our algorithm of UniPMatMed. Since cost analysis for the filtering stage only uses ϕ (and not λ), our analysis for cost will be identical to that of our analysis of UniPMatMed, therefore we will have a cost guarantee of 8.

To analyze the radius guarantee, notice that while λ does not explicitly use radius values, PMatMed LP has a constraint that ensures $\forall j \in \mathcal{C}$ $\bar{C}_j \leq r_j$. Therefore, $\lambda(j) = 2\bar{C}_j \leq 2r_j$. Our initial setting of λ ($\lambda(j) := \min\{r_j, 2\bar{C}_j\}$) made it so $\lambda(j) \leq r_j$. Hence, the new setting of λ will worsen the radius guarantee of the final solution. The analysis for this result can be found in Appendix C.

4.3 Tighter radius guarantee for UniPMatMed

In the previous result for UniPMatMed, we set $\phi(j) := \bar{C}_j$ and $\lambda := \min\{L, 2\bar{C}_j\}$. In the second result for PMatMed, we showed how setting $\lambda(j) := 2L$ would increase the radius guarantee. Thus, in order to tighten the radius guarantee for UniPMatMed, we will again change $\lambda(j)$, but this time in a way that will allow points to have tighter radius bounds.

To build up to our new setting for λ, we first partition points in the original client set into points that have relatively small, or *tiny* $\bar{C}_j$ values, $\mathcal{C}_T = \{j \in \mathcal{C} \mid \bar{C}_j \leq \epsilon L\}$ and points that have *large* $\bar{C}_j$ values, $\mathcal{C}_L = \{j \in \mathcal{C} \mid \bar{C}_j > \epsilon L\}$. Now, our algorithm is as follows: Run Algorithm 2 on PMatMed instance $\mathscr{I}$, but in Line 0, set $\phi(j) := \bar{C}_j$ and $\lambda(j)$ as defined below.

$$\lambda(j) = \begin{cases} 2\bar{C}_j & j \in \mathcal{C}_T \\ L & j \in \mathcal{C}_L \end{cases}$$

Note that ϕ and λ will satisfy compatibility. Furthermore, this setting of λ improves the radius bound of Section 4.1 since it forces cluster centers from $\mathcal{C}_L$ to open their own primary and secondary facilities (i.e. they will force $\sigma(j) = j$ for cluster centers j that are from $\mathcal{C}_L$). Any point j such that $\sigma(j) \neq j$ will be from $\mathcal{C}_T$, and furthermore $\sigma(j) \in \mathcal{C}_T$ for all of these points. Therefore, we are decreasing the distance between any point and its secondary facility, for both points in $\mathcal{C}_L$ and $\mathcal{C}_T$. As noted in Remark 27, this will help reduce the radius guarantee.

To achieve the $(5+8\epsilon, 4+2/\epsilon)$-approximation result, we also make a change to Section 2.4 of Filter. Details about this change, as well as the full analysis for this result can be found in the full version of this paper.

References

1 Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 504–514, 2021.

2 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4):FOCS17–97–FOCS17–156, 2020. doi:10.1137/18M1171321.

3 Soroush Alamdari and David B. Shmoys. A bicriteria approximation algorithm for the k-center and k-median problems. In *WAOA*, 2017.

4 Ali Aouad and Danny Segev. The ordered k-median problem: surrogate models and approximation algorithms. *Mathematical Programming*, 177(1):55–83, 2019.

5 David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

6 Tanvi Bajpai, Deeparnab Chakrabarty, Chandra Chekuri, and Maryam Negahbani. Revisiting priority k-center: Fairness and outliers. *arXiv preprint*, 2021. arXiv:2103.03337.

7 Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi Varadarajan. A Constant Approximation for Colorful k-Center. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ESA.2019.12.

8 Jarosław Byrka, Krzysztof Sornat, and Joachim Spoerhase. Constant-factor approximation for ordered k-median. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 620–631, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188930.

9 Deeparnab Chakrabarty and Maryam Negahbani. Better algorithms for individually fair k-clustering, 2021. arXiv:2106.12150.

10 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002. doi:10.1006/jcss.2002.1882.

11 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30, 2017.

12 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink!*, pages 11–26. Springer, 2003.

13 Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 438–448, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3442188.3445906.

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

15 Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10(2):180–184, May 1985. doi:10.1287/moor.10.2.180.

16 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, March 2001. doi:10.1145/375827.375845.

17 Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location. *arXiv preprint*, 2019. arXiv:1908.09041.

18 Naoyuki Kamiyama. The distance-constrained matroid median problem. *Algorithmica*, 82(7):2087–2106, July 2020. doi:10.1007/s00453-020-00688-5.

19 T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002. doi:10.1109/TPAMI.2002.1017616.

20 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *International Conference on Machine Learning*, pages 3448–3457. PMLR, 2019.

21 Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1117–1130, USA, 2011. Society for Industrial and Applied Mathematics.

22 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*, pages 646–659, 2018.

23 Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering, 2020. arXiv:2002.06742.

24 Stefan Nickel and Justo Puerto. *Location theory: a unified approach*. Springer Science & Business Media, 2006.

25 J. Plesník. A heuristic for the p-center problems in graphs. *Discrete Applied Mathematics*, 17(3):263–268, 1987. doi:10.1016/0166-218X(87)90029-1.

26 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

27 Chaitanya Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. *ACM Trans. Algorithms*, 12(4), August 2016. doi:10.1145/2963170.

28 Ali Vakilian and Mustafa Yalçıner. Improved approximation algorithms for individually fair clustering, 2021. doi:10.48550/arXiv.2106.14043.

A Omitted Proofs

Proof of Lemma 10. We first prove that $COST'(x,y) \leq 2 \cdot COST(x,y)$. The fractional facility opening cost, $\sum_i f_i y_i$ is identical in both. The difference in the client connection cost is because the demands of clients in $\mathcal{C} \setminus C$ are relocated. Consider a client $k \in \mathcal{C} \setminus C$ that is relocated to its cluster center $j \in C$ (thus $k \in D(j)$). In $COST(x,y)$ client k pays $a_k \bar{C}_k$. In $COST'(x,y)$, the demand of k is moved to j and pays $a_k \bar{C}_j$. Thus, it suffices to prove that $\bar{C}_j \leq 2\bar{C}_k$. From Fact 9, $\phi(j) \leq \phi(k) \leq 2\bar{C}_k$. LP constraints 1d and 1c of the LP for $\mathscr{I}$ ensures that $\bar{C}_j \le r_j$ for all $j \in \mathcal{C}$. Hence, if $\bar{C}_j > 2\bar{C}_k$ we would have $\phi(j) = \min\{r_j, 2\bar{C}_j\} > 2\bar{C}_k$ which would be a contradiction to $\phi(j) \le \phi(k)$. This shows that $\bar{C}_j \le 2\bar{C}_k$.

Now we consider the second part. From Fact 9, $d(j,k) \le 2\lambda(k) \le 2(2 \cdot \bar{C}_k)$. Suppose the cost of an integer solution to $\mathscr{I}'$ is α. We keep the same facilities for $\mathscr{I}$ and account for the increase in connection cost when considering the original client locations. Consider a client $k \in \mathcal{C} \setminus C$ that is relocated to center $j \in C$. If j connects to i in the integer solution for $\mathscr{I}'$, k can connect to i in the solution to $\mathscr{I}$, and its per unit connection cost increases by at most $d(j,k) \le 4\bar{C}_k$. Thus the total increase in the connection cost when comparing to α is upper bounded by $\sum_{j \in C} \sum_{k \in D(j)} a_k \cdot 4\bar{C}_k \le 4 \cdot COST(x,y)$. ◀

Proof of Lemma 13. Suppose $\mathcal{Q}$ is non-empty and v^* is any extreme point. Then v^* is the unique solution of a linear system $Av = b$ where A is a subset of the inequalities of $\mathcal{Q}$ with A having full row and column rank (in particular the rows of A are linearly independent vectors). A can be partitioned into A_1 and A_2 where A_1 is a subset of the inequalities coming from the matroid $\mathcal{M}$ (of the form $v(S) = r_{\mathcal{M}}(S)$), while A_2 is a subset of the remaining inequalities. Via the submodularity of the matroid rank function, it is known that one can choose A_1 such that the rows of A_1 correspond to a laminar family of subsets of $\mathcal{F}$ [26]. We observe that the non-matroidal system of inequalities in $\mathcal{Q}$ correspond to a laminar family of sets over $\mathcal{F}$: (a) the sets $G_j, j \in C$ are disjoint and $F'_j \subseteq G_j$ for each j and (b) for $j \in C_s$, we have $B_j \subseteq G_j$. See Figure 1.

Thus the rows of the matrix of A come from two laminar families of sets over $\mathcal{F}$, and it is known that such a matrix is totally uniodular [26]. Thus $v^* = A^{-1}b$ where A^{-1} is an integer matrix, and b is half-integral which implies that v^* is half-integral. ◀

Proof of Lemma 17. We first show that $T(y') \le 4 \cdot COST'(x,y)$ (we already have $T(\hat{y}) \le T(y')$). We know that $COST'(x,y)$ can be expressed as $\sum_i f_i y_i + \sum_j a'_j \cdot \bar{C}_j$. For any $j \in C$, observe that $\bar{C}_j = \sum_{i \in G_j} d(i,j)x_{ij} + \sum_{i \notin G_j} d(i,j)x_{ij}$ and hence $\bar{C}_j \ge \sum_{i \in G_j} d(i,j)x_{ij} + \gamma_j \sum_{i \notin G_j} x_{ij}$.

$$\begin{aligned} T(y') &\le \sum_i f_i y_i + \sum_j a'_j \Big(2 \sum_{i \in G_j} d(i,j)x_{ij} + 4\gamma_j \Big(1 - \sum_{i \in G_j} x_{ij}\Big)\Big) \\ &\le \sum_i f_i y_i + 4 \sum_j a'_j \cdot \bar{C}_j \le 4 \cdot COST'(x,y) \end{aligned}$$

Next, we upper bound $COST'(\hat{x},\hat{y})$ by $T(\hat{y})$. It suffices to focus on the assignment cost. Consider $j \in C_s$. Its primary and secondary facilities are in B_j and it is easy to see that its connection cost is precisely $\sum_{i \in B_j} d(i,j)\hat{x}_{ij}$. Now consider $j \in C_b$. Recall that when $\hat{y}(G_j) = 1$, the total assignment cost of j is at most $\sum_{i \in G_j} d(i,j)\hat{y}_i$. When $\hat{y}(G_j) < 1$, j connects to primary facility in F'_j and a secondary facility. The second nearest facility will not be in its G_j ball, i.e. $i_2(j) \notin F_j$. Let $j' \neq j$ be client that defines γ_j. Via Lemma 14, we have $d(i_2(j), j) \le 3\gamma_j$. Assuming this, when $\hat{y}(G_j) < 1$, the total assignment cost of j is at most $\sum_{i \in G_j} d(i,j)\hat{y}_i + 3\gamma_j(1 - \hat{y}(G_j))$. Based on these assignment cost upper bounds we see that $COST'(\hat{x},\hat{y}) \le T(\hat{y})$.

Now we prove Lemma 14. From Fact 9 we have $2\max\{\lambda(j), \lambda(j')\} \le d(j,j')$. Via triangle inequality $d(j,j') \le d(j,i) + d(i,j') \le 2\gamma_j$. Thus $2\lambda(j') \le 2\gamma_j$ which implies that $\lambda(j') \le \gamma_j$. Recall that $F'_{j'}$, from its definition, is contained in a ball of radius $\lambda(j')$ around j'. Thus, for any facility $i' \in F'_{j'}$, $d(i',j') \le \lambda(j') \le \gamma_j$, Therefore, $d(i',j) \le d(j,j') + d(j',i') \le 3\gamma_j$. This gives us the lemma.

Finally, using Lemma 10, we know that $COST'(x,y) \le 2 \cdot COST(x,y)$, hence $4 \cdot COST'(x,y) \le 8 \cdot COST(x,y)$. ◀

Proof of Lemma 22. Since the facility costs of $(\tilde{x}, \tilde{y})$ will remain as they are in $H(\tilde{y})$, it suffices to show that for all $j \in C$, the assignment cost of j is at most $L_j(\tilde{y})$. When $j \in C'$, $\mathsf{ctr}(j) = j$ and the assignment cost of j will be exactly $L_j(\tilde{y})$.

Now we consider two possibilities for $j' \in C \setminus C'$. Let $\mathsf{ctr}(j') = j$. If j' gets assigned to a center from S_j, there are two possible cases for the value of $L_{j'}(\tilde{y})$. If $i_1(j') \in S_j$ then the assignment cost for j' is exactly $L_{j'}(\tilde{y})$. Otherwise, $i_1(j') \notin S_j$ and $\tilde{y}_{i_1(j')} = 0$. In this case $L_{j'}(\tilde{y}) = \sum_{i \in S_j} a'_{j'}(d(j', \sigma(j)) + d(i, \sigma(j)))\tilde{y}_i$. By triangle inequality, $d(i, j') \leq d(i, \sigma(j')) + d(j, \sigma(j'))$, therefore the assignment cost of j' is at most $L_{j'}(\tilde{y})$.

If j' is assigned to a center that is not from S_j, it is because $\tilde{y}_{i_1(j')} = 1$ and $i_1(j') \notin S_j$. Here, the assignment cost of j' is $a'_{j'}d(i_1(j'), j')$. Let $i \in S_j$ be such that $\tilde{y}_i = 1$. The value of $L_{j'}(\tilde{y})$ is therefore

$$\begin{aligned} L_{j'}(\tilde{y}) &= a'_j\Big(d(j', \sigma(j')) + d(i, \sigma(j')) + d(i_1(j'), j) - d(j', \sigma(j')) - d(i_1(\sigma(j)), \sigma(j))\Big) \\ &= a'_j\Big(d(i, \sigma(j')) + d(i, i_1(j')) - d(i_1(\sigma(j)), \sigma(j))\Big) \end{aligned}$$

Since $i \in S_j$ cannot be closer to $\sigma(j')$ than the primary facility of $\sigma(j')$, we know that $d(i_1(\sigma(j)), \sigma(j)) \leq d(i, \sigma(j'))$. Thus, the assignment cost of j' is at most $L_{j'}(\tilde{y})$. ◀

Proof of Lemma 23. For notational ease, let $Q_j(\hat{y}) := 2\sum_{i \in G_j} d(i, j)\hat{y}_i + 4\gamma_j(1 - \sum_{i \in G_j} \hat{y}_i)$. Thus, $T(\hat{y}) = \sum_i f_i\hat{y}_i + \sum_{j \in C} a'_j Q_j(\hat{y})$. As in the proof of the previous lemma, we focus on just the assignment costs of clients, since clearly $\sum_i f_i\hat{y}'_i \leq \sum_i f_i\hat{y}_i$. Specifically, we will show that $L_j(\hat{y}') \leq a'_j Q_j(\hat{y})$ for all $j \in C$. For the remainder of the proof, we omit the term a'_j from both sides of this inequality, since it remains fixed throughout our analysis.

First, we show $\hat{C}_j \leq Q_j(\hat{y})$ for all $j \in C$. Recall that j has no secondary facility when $\hat{y}_{i_1(j)} = 1$, in which case $i_2(j) = i_1(j)$. When $\hat{y}(G_j) = 1$, $\sigma(j) = j$ and the primary and secondary facilities of j are the only facilities in G_j where $\hat{y}_i > 0$. Since $\hat{y}$ is half integral, we get $\hat{C}_j = (d(i_1(j), j) + d(i_2(j), j))/2 = \sum_{i \in G_j} d(i, j)\hat{y}_i \leq Q_j(\hat{y})$. When $\hat{y}(G_j) = 1/2$, $\sigma(j) = \ell \neq j$ and $i_2(j) = i_1(\ell)$. In this case $\hat{C}_j = (d(i_1(j), j) + d(j, \ell) + d(i_1(\ell), \ell))/2$. Using Claim 15 and definitions, $d(j, \ell) + d(\ell, i_1(\ell)) \leq 3\gamma_j$. Therefore $\hat{C}_j \leq \sum_{i \in G_j} d(i, j)\hat{y}_i + 3\gamma_j(1 - \hat{y}(G_j)) \leq Q_j(\hat{y})$. To prove $L_j(\hat{y}') \leq a'_j Q_j(\hat{y})$ we consider several cases.

1. $j \in C'$: we have $\mathsf{ctr}(j) = j$ and $i_1(j) \in S_j$.

$$\begin{aligned} L_j(\hat{y}') = \sum_{i \in S_j} d(i, j)\hat{y}'_i \leq \sum_{i \in S_j} d(i, j)\hat{y}_i &= \frac{1}{2}(\Big(d(i_1(j), j) + d(i_2(j), j)\Big) \\ &\leq \frac{1}{2}\Big(d(i_1(j), j) + d(i_1(j), \sigma(j)) + d(i_2(j), \sigma(j))\Big) \quad \text{(via triangle ineq.)} \\ &= \hat{C}_j \leq Q_j(\hat{y}). \end{aligned}$$

2. $j' \in C \setminus C'$. Let $\mathsf{ctr}(j') = j$. We have $\hat{C}_j \leq \hat{C}_{j'}$.
   a. $i_1(j') \in S_j$. Then $i_2(j) = i_1(j')$ hence $\sigma(j) = j'$.

$$\begin{aligned} L_{j'}(\hat{y}') &= \frac{1}{2}(d(i_1(j), j') + d(i_2(j), j')) \\ &\leq \frac{1}{2}(d(i_1(j), j) + d(j, j') + d(i_2(j), j')) \quad \text{(via triangle ineq.)} \\ &= \hat{C}_j \leq \hat{C}_{j'} \leq Q_{j'}(\hat{y}). \end{aligned}$$

   b. $i_1(j') \notin S_j$: Then $S_j \cap S_{j'}$ is either $\{i_1(j)\}$ or $\{i_2(j)\}$ (Claim 21). In both cases, $\sigma(j') = \ell \neq j'$ and therefore $\hat{y}(G_{j'}) = \hat{y}_{i_1(j')} = 1/2$. Hence

$$L_{j'}(\hat{y}') = \frac{1}{2} \cdot \Big(2d(j', \ell) + d(i_1(j), \ell) + d(i_2(j), \ell) + d(i_1(j'), j') - d(j', \ell) - d(i_1(\ell), \ell)\Big)$$

i. When $S_j \cap S_{j'} = \{i_1(j)\}$, $i_1(j) = i_2(j')$ thus $\ell = j$. Using the fact that $d(i_2(j), j) \leq 2\hat{C}_j - d(i_1(j), j)$, we have

$$\begin{aligned}
L_{j'}(\hat{y}') &= \frac{1}{2}\Big(2d(j', j) + d(i_1(j), j) + d(i_2(j), j) + d(i_1(j'), j') - d(j', j) - d(i_1(j), j)\Big) \\
&= \frac{1}{2}\Big(d(j, j') + d(i_2(j), j) + d(i_1(j'), j')\Big) \\
&\leq \frac{1}{2}\Big(d(j, j') + 2\hat{C}_j - d(i_1(j), j) + d(i_1(j'), j')\Big) \\
&\leq \frac{1}{2}\Big(d(j, j') + 2\hat{C}_{j'} - d(i_1(j), j) + d(i_1(j'), j')\Big) \\
&= d(j, j') + d(i_1(j'), j').
\end{aligned}$$

ii. When $S_j \cap S_{j'} = \{i_2(j)\}$, $i_2(j) = i_2(j') = i_1(\ell)$ and so $\ell \neq j, j'$ and $\sigma(j) = \sigma(j') = \ell$. Since $2\hat{C}_j \leq 2\hat{C}_{j'}$, $d(i_1(j), j) + d(j, \ell) \leq d(i_1(j'), j') + d(j', \ell)$. Therefore,

$$\begin{aligned}
L_{j'}(\hat{y}') &= \frac{1}{2}\Big(d(j', \ell) + d(i_1(j), \ell) + d(i_1(j'), j')\Big) \\
&\leq \frac{1}{2}\Big(d(j', \ell) + d(i_1(j), j) + d(j, \ell) + d(i_1(j'), j')\Big) \quad \text{(via triangle ineq.)} \\
&\leq \frac{1}{2}\Big(d(j', \ell) + d(i_1(j'), j') + d(j', \ell) + d(i_1(j'), j')\Big) \\
&\leq d(i_1(j'), j') + d(j', \ell).
\end{aligned}$$

Thus, in both cases we have

$$\begin{aligned}
L_{j'}(\hat{y}') &\leq d(i_1(j'), j') + d(j', \ell) \\
&\leq d(i_1(j'), j') + 2\gamma_{j'} \quad \text{(via Claim 15)} \\
&\leq 2\sum_{i \in G_{j'}} d(i, j')\hat{y}_i + 4\gamma_{j'}(1 - \sum_{i \in G_{j'}} \hat{y}_i) = Q_{j'}(\hat{y}) \quad (\text{since } \hat{y}(G_{j'}) = 1/2).
\end{aligned}$$

This finishes the case analysis and the proof. ◀

B Uniform Priority Matroid Median

The UniPMatMed problem is a special case of the PMatMed problem in which all clients have the same radius value L. An instance $\mathscr{J}$ of the UniPMatMed problem can be described using the tuple $(\mathcal{F}, \mathcal{C}, d, \mathbf{f}, L, \mathbf{a}, \mathcal{M})$. We will abuse notation and interpret L as not only a single radius value, but also as a vector from $\mathbb{R}^{\mathcal{C}}$ where each entry is L; this will allow us to use our algorithm for PMatMed on instances of UniPMatMed.

In this section we show how we can take advantage of the uniform radius requirement to improve upon the $(21, 12)$-approximation for PMatMed. In particular, since we have $\bar{C}_j \leq L$ for all $j \in \mathcal{C}$, we can pick points in filtering in order of their $\bar{C}_j$ values and set $\phi(j) := \bar{C}_j$ for Filter. This setting of ϕ will be compatible with the setting of $\lambda(j) := \min\{L, 2\bar{C}_j\}$. Furthermore, Filter with these ϕ and λ functions is identical to the filtering step in Kamiyama's algorithm [18]. Notice that these same settings for PMatMed, i.e. $\phi(j) := \bar{C}_j$ and $\lambda := \min\{r_j, 2\bar{C}_j\}$, are not necessarily compatible. The uniform radius constraint also help us to derive tighter bounds throughout the radius analysis of the PMatMed algorithm.

Using the above observations, our algorithm for UniPMatMed is the following: Run Algorithm 2 on UniPMatMed instance $\mathscr{J}$, but in Line 0, set $\phi(j) := \bar{C}_j$ and $\lambda(j) := \min\{r_j, 2\bar{C}_j\}$. Thus, the only change in the algorithm is the filtering step. We argue that this algorithm yields a better approximation algorithm for UniPMatMed.

▶ **Theorem 29** (Theorem 2a). *There is a $(9, 8)$-approximation algorithm for UniPMatMed.*

B.1 Cost and Radius Analysis for UniPMatMed

Since our algorithm for UniPMatMed only slightly differs from the one in Section 3, we omit several proofs that would be identical. The only change to the cost analysis occurs in the filtering stage (Section 3.1). In particular, we can derive a tighter bound than in Lemma 10. This ultimately leads to the improved cost bound, shown in Theorem 31.

▶ **Lemma 30.** *The following is true of $\mathscr{J}'$. (a) $COST'(x,y) \leq COST(x,y)$. (b) Any integer solution (x',y') for $\mathscr{J}'$ can be converted to an integer solution for $\mathscr{J}$ that incurs an additional cost of at most $4 \cdot COST(x,y)$.*

Recall that $(\tilde{x},\tilde{y})$ is the final integeral solution output by the algorithm.

▶ **Theorem 31.** *$COST(\tilde{x},\tilde{y}) \leq 8 \cdot COST(x,y)$.*

We now analyze the radius guarantee and outline the changes in the analysis. First, we have the following lemma in place of Lemma 11 which also follows directly from Fact 9.

▶ **Lemma 32.** *Let $k \in \mathcal{C}$ be assigned to $j \in C$ after using the Filtering procedure (i.e. $k \in D(j)$). Then, $d(j,k) \leq 2\lambda(k) \leq 2L$.*

Since all radius values are equal, we do not need Lemma 18 to relate the radius values of different clients. We do need to update Lemma 19 and Lemma 26. These updated lemmas are given below.

▶ **Lemma 33.** *Let $k \in \mathcal{C}$ and $k \in D(j)$ for a cluster center $j \in C$. Then (a) $d(j,i_1(j)) \leq \lambda(j) \leq \lambda(k) \leq L$ and (b) when $j \in C_s$ $d(j,i_2(j)) \leq \rho_j \leq L$ and (c) when $j \in C_b$ $d(j,i_2(j)) \leq d(j,\sigma(j)) + d(i_1(\sigma(j)),\sigma(j)) \leq 3\gamma_j \leq 3\rho_j \leq 3L$.*

The reasoning for the preceding lemma is the same as Lemma 19, except L is used in place of r_j and r_k values. This results in the following update to Lemma 26

▶ **Lemma 34.** *Let $j \in C$. The final solution will open a facility i such that $d(i,j) \leq 7L$.*

Lemma 32 and Lemma 34 give us the following improved radius bound for the solution output by the algorithm. This, along with Theorem 31, proves Theorem 29.

▶ **Theorem 35.** *Let S be the output of the aforementioned approximation algorithm for UniPMatMed. For all $k \in \mathcal{C}$, $d(k,S) \leq 9L$.*

C Analysis for $(36,8)$-approximation for PMatMed

In this section we show how to obtain a $(36,8)$-approximate solution for PMatMed. Our algorithm is as follows: Run Algorithm 2 on PMatMed instance $\mathscr{I}$, but in Line 0, set $\phi(j) := \bar{C}_j$ and $\lambda(j) := 2\bar{C}_j$. Clearly, ϕ and λ are compatible. Furthermore, notice that this setting of ϕ is identical to that of our algorithm of UniPMatMed. Since cost analysis for the filtering stage of UniPMatMed only uses ϕ (and not λ), Lemma 30 and Theorem 31 hold in this case as well. This is the reason why the cost factor guarantee will be 8.

Though our setting for λ does not use radius values, from the PMatMed LP constraint, $\forall j \in \mathcal{C}$, $\bar{C}_j \leq r_j$ holds. Therefore, $\lambda(j) = 2\bar{C}_j \leq 2r_j$. Previous settings of λ (where $\lambda(j) := \min\{r_j, 2\bar{C}_j\}$) made it so $\lambda(j) \leq r_j$. Thus the new setting of λ can lead to a weakening of the radius guarantee. First, we formalize the above observation in Fact 36 which we will use to update the radius analysis of Section 3.

▶ **Fact 36.** *The following holds after* Filter *when* $\phi(j) := \bar{C}_j$ *and* $\lambda(j) := 2\bar{C}_j$*: (a)* $\bar{C}_j \leq r_j$*, and hence* $\lambda(j) = 2\bar{C}_j \leq 2r_j$*, (b)* $\forall k \in D(j) \quad d(j,k) \leq 2\lambda(k) \leq 4C_k \leq 4r_k$*.*

The following updated lemmas now hold in place of their counterparts from Section 3. The proofs for these results are identical to those from Section 3 up to certain bounds that change due to the above fact and the subsequent lemmas. These changes occur whenever definitions of ϕ and λ are used in the analysis, and the following lemmas will be invoked in place of their counterparts from Section 3.

▶ **Lemma 37** (Updated Lemma 11)**.** *Let* $k \in \mathcal{C}$ *be assigned to* $j \in C$ *after using the Filtering procedure (i.e.* $k \in D(j)$*). Then,* $d(j,k) \leq 2\lambda(k) \leq 4r_k$*.*

▶ **Lemma 38** (Updated Lemma 18)**.** *For some* $k \in \mathcal{C}$*, where* $k \in D(j)$*,* $\rho_j \leq 5r_k$*.*

▶ **Lemma 39** (Updated Lemma 19)**.** *Let* $k \in \mathcal{C}$ *where* $k \in D(j)$ *for* $j \in C$*. (a)* $d(j, i_1(j)) \leq \lambda(j) \leq \lambda(k) \leq 2r_k$*, (b) when* $j \in C_s$*,* $d(j, i_2(j)) \leq \rho_j \leq 5r_k$*, and (c) when* $j \in C_b$*,* $d(j, i_2(j)) \leq d(j, \sigma(j)) + d(i_1(\sigma(j)), \sigma(j)) \leq 3\gamma_j \leq 3\rho_j \leq 15r_k$*.*

▶ **Lemma 40** (Updated Lemma 26)**.** *Let* $k \in \mathcal{C}$*, where* $k \in D(j)$ *for* $j \in C$*. The final solution will open a facility* i *such that* $d(i,j) \leq 32r_k$*.*

Finally, using Lemma 37 Lemma 40, along with Theorem 31, we get the following result.

▶ **Theorem 41** (Theorem 1(b))**.** *There is a* $(36, 8)$*-approximation algorithm for Priority Matroid Median.*