Improved Approximation Algorithms for Matroid and Knapsack Median Problems and Applications[*]

Chaitanya Swamy

Combinatorics and Optimization, University Waterloo, Waterloo, ON N2L 3G1
cswamy@math.uwaterloo.ca

Abstract

We consider the *matroid median* problem [11], wherein we are given a set of facilities with opening costs and a matroid on the facility-set, and clients with demands and connection costs, and we seek to open an independent set of facilities and assign clients to open facilities so as to minimize the sum of the facility-opening and client-connection costs. We give a simple 8-approximation algorithm for this problem based on LP-rounding, which improves upon the 16-approximation in [11]. We illustrate the power and versatility of our techniques by deriving: (a) an 8-approximation for the *two-matroid median* problem, a generalization of matroid median that we introduce involving two matroids; and (b) a 24-approximation algorithm for *matroid median with penalties*, which is a vast improvement over the 360-approximation obtained in [11]. We show that a variety of seemingly disparate facility-location problems considered in the literature—data placement problem, mobile facility location, k-median forest, metric uniform minimum-latency UFL—in fact reduce to the matroid median or two-matroid median problems, and thus obtain *improved* approximation guarantees for all these problems. Our techniques also yield an improvement for the knapsack median problem.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2 Discrete Mathematics

Keywords and phrases Approximation algorithms, LP rounding, facility location, matroid and submodular polyhedra, knapsack constraints

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2014.403

1 Introduction

We investigate facility location problems wherein the set of open facilities have to satisfy some matroid independence constraints or knapsack constraints. Specifically, we consider the *matroid median problem*, which is defined as follows. As in the uncapacitated facility location problem, we are given a set of facilities $\mathcal{F}$ and a set of clients $\mathcal{D}$. Each facility i has an *opening cost* of f_i. Each client $j \in \mathcal{D}$ has demand d_j and assigning client j to facility i incurs an *assignment cost* of $d_j c_{ij}$ proportional to the distance between i and j. Further, we are given a matroid $M = (\mathcal{F}, \mathcal{I})$ on the set of facilities. The goal is to choose a set $F \in \mathcal{I}$ of facilities to open that forms an independent set in M, and assign each client j to a facility $i(j) \in F$ so as to minimize the total facility-opening and client-assignment costs, that is, $\sum_{i \in F} f_i + \sum_{j \in \mathcal{D}} d_j c_{i(j)j}$. We assume that the facilities and clients are located in a common metric space, so the distances c_{ij} form a metric.

[*] A full version is available on the CS arXiv, see http://arxiv.org/abs/1310.7834.
This work was supported in part by NSERC grant 327620-09, an NSERC Discovery Accelerator Supplement Award, and an Ontario Early Researcher Award.

17th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14) /
18th Int'l Workshop on Randomization and Computation (RANDOM'14).
Editors: Klaus Jansen, José Rolim, Nikhil Devanur, and Cristopher Moore; pp. 403–418
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The matroid median problem generalizes the metric *k-median* problem, which is the special case where M is a uniform matroid (and there are no facility-opening costs), and is thus, *NP*-hard. The matroid median problem without facility-opening costs was introduced recently by Krishnaswamy et al. [11], who gave a 16-approximation algorithm for this problem.

Our contributions are threefold.

- We devise an improved 8-approximation algorithm for the matroid-median problem (Section 3). Moreover, notably, our algorithm is significantly simpler and cleaner than the one in [11], and satisfies the stronger property that it is a *Lagrangian-multiplier-preserving* 8-approximation algorithm (see Remark 3.4). The effectiveness and versatility of our simpler approach for matroid median is further highlighted when we consider some natural extensions of matroid median in Section 4. We leverage the techniques underlying our simpler and cleaner algorithm for matroid median to devise: (a) an 8-approximation algorithm for the *two-matroid median* problem (Section 4.1), which is an extension that we introduce involving two matroids that captures some interesting facility-location problems considered in the literature; and (b) a 24-approximation algorithm (Section 4.2) for the *matroid median problem with penalties*, wherein we are allowed to leave client unassigned and incur a penalty for each unassigned client; this constitutes a vast improvement over the approximation ratio of 360 obtained by Krishnaswamy et al. [11].
- We show that the matroid median and two-matroid median problem turn out to be rather fundamental problems by showing in Section 5 that a variety of facility location problems that have been considered in the literature can be cast as instances of matroid median or two-matroid median. These include the data placement problem [2, 3], mobile facility location [9, 1], k-median forest [10], and metric uniform minimum-latency UFL [4]. This not only gives a unified framework for viewing these seemingly disparate problems, but also our approximation guarantee of 8 *yields improved, and in some cases, the first, approximation guarantees for all these problems.*
- We adapt our techniques to also obtain an improvement for the knapsack median problem [11, 12] (Section 6).

Our improvement for matroid median comes from an improved, simpler rounding procedure for a natural LP relaxation of the problem also considered in [11]. We show that a clustering step introduced in [5] for the k-median problem coupled with two applications of the integrality of the intersection of two submodular (or matroid) polyhedra—one to obtain a half-integral solution, and another to obtain an integral solution—suffices to obtain the desired approximation ratio. In contrast, the algorithm in [11] starts off with the clustering step in [5], but then further dovetails the rounding procedure of [5] creating trees, then stars, and then applies the integrality of the intersection of two submodular polyhedra.

There is great deal of similarity between the the rounding algorithm of [11] for matroid median and the rounding algorithm of Baev and Rajaraman [2] for the data placement problem, who also perform the initial clustering step in [5] and then create trees and then stars and use these to obtain an integral solution. In contrast, our simpler, improved rounding algorithm is similar to the rounding algorithm in [3] for data placement, who use the initial clustering step of [5] coupled with two min-cost flow computations—one to obtain a half-integral solution and another to obtain an integral solution—to obtain the final solution. These similarities are not surprising since, as mentioned above, we show in Section 5 that the data-placement problem is a special case of the matroid median problem. In fact, our improvements are analogous to those obtained for the data-placement problem by Baev, Rajaraman, and Swamy [3] over the guarantees in [2], and stem from similar insights.

A common theme to emerge from our work and [3] is that in various settings, the initial clustering step introduced by [5] imparts sufficient structure to the fractional solution so that one can then round it using two applications of suitable integrality-results from combinatorial optimization. First, this initial clustering can be used to derive a half-integral solution. This was observed explicitly in [2] and is implicit in [11], and making this explicit yields significant dividends. Second, and this is the oft-overlooked insight (in [2, 11]), a half-integral solution can be easily rounded, and in a better way, *without resorting to creating trees and then stars etc. as in the algorithm of [5].* This is due to the fact that a half-integral solution is already "filtered": if client j is assigned to facility i fractionally, then one can bound c_{ij} in terms of the assignment cost paid by the fractional solution for j (see Section 3). This enables one to use a standard facility-location clustering step to set up a suitable combinatorial-optimization problem possessing an integrality property, and hence, round the half-integral solution. The resulting algorithm is typically both simpler and has a better approximation ratio than what one would obtain by mimicking the steps of [5] involving creating trees, stars etc.

Recently, Charikar and Li [6] obtained a 9-approximation algorithm for the matroid-median problem; our results were obtained independently. While there is some similarity between our ideas and those in [6], we feel that our algorithm and analysis provides a more illuminating explanation of why matroid median and some of its extensions (e.g., two-matroid median, matroid median with penalties; see Section 4) are "easy" to approximate, whereas other variants such as matroid-intersection median (Section 4) are inapproximable. It is possible that our ideas coupled with the dependent-rounding procedure used in [6] for the k-median problem may lead to further improvements for the matroid median problem; we leave this as future work.

2 An LP Relaxation for Matroid Median

We can express the matroid median problem as an integer program and relax the integrality constraints to get an LP. Throughout we use i to index facilities in $\mathcal{F}$, and j to index clients in $\mathcal{D}$. Let r denote the rank function of the matroid $M = (\mathcal{F}, \mathcal{I})$.

$$\begin{aligned} \min \quad & \sum_i f_i y_i + \sum_j \sum_i d_j c_{ij} x_{ij} && && \text{(P)} \\ \text{s.t.} \quad & \sum_i x_{ij} = 1 && \forall j \\ & \sum_{i \in S} y_i \leq r(S) && \forall S \subseteq \mathcal{F} \\ & 0 \leq x_{ij} \leq y_i && \forall i, j. \end{aligned}$$

Variable y_i indicates if facility i is open, and x_{ij} indicates if client j is assigned to facility i. The first and second constraints say that each client must be assigned to an open facility. The third constraint encodes the matroid independence constraint. An integer solution corresponds exactly to a solution to our problem. We note that (P) can be solved in polytime since (for example) a polytime algorithm for submodular-function minimization yields an efficient separation oracle.

3 A Simple 8-Approximation Algorithm via LP-Rounding

Let (x, y) denote an optimal solution to (P) and OPT be its value. We first describe a simple algorithm to round (x, y) to an integer solution losing a factor of at most 10. In Section 3.4, we use some additional insights to improve the approximation ratio to 8. We use the terms connection cost and assignment cost interchangeably.

3.1 Overview of the Algorithm

We first give a high level description of the algorithm. Suppose for a moment that the optimal solution (x, y) satisfies the following property:

$$\text{for every facility } i\text{, there is } \textit{at most one} \text{ client } j \text{ such that } x_{ij} > 0. \tag{$*$}$$

Let $\mathcal{F}_j = \{i : x_{ij} > 0\}$. Notice that the $\mathcal{F}_j$ sets are disjoint. We may assume that for $i \in \mathcal{F}_j$, we have $y_i = x_{ij}$, so the objective function is a linear function of only the y_i variables. We can then set up the following matroid intersection problem. The first matroid is M restricted to $\bigcup_j \mathcal{F}_j$. The second matroid M' (on the same ground set $\bigcup_j \mathcal{F}_j$) is the partition matroid defined by the $\mathcal{F}_j$ sets; that is, a set is independent in M' if it contains at most one facility from each $\mathcal{F}_j$. Notice the y_i-variables yield a fractional point in the *intersection of the matroid polyhedron of M and the matroid-base polyhedron of M'*. Since the intersection of these two polyhedra is known to be integral (see, e.g., [8]), this means that we can round (x, y) to an integer solution of no greater cost. Of course, the LP solution need not have property $(*)$ so our goal will be to transform (x, y) to a solution that has this property without increasing the cost by much.

Roughly speaking we want to do the following: cluster the clients in $\mathcal{D}$ around certain 'centers' (also clients) such that (a) every client k is assigned to a "nearby" cluster center j whose LP assignment cost is less than that of k, and (b) the facilities serving the cluster centers in the fractional solution (x, y) are disjoint. So, the modified instance where the demand of a client is moved to the center of its cluster has a fractional solution, namely the solution induced by (x, y), that satisfies $(*)$ and has cost at most OPT. Furthermore, given a solution to the modified instance we can obtain a solution to the original instance losing a small additive factor. One option is to use the decomposition method of Shmoys et al. [13] for uncapacitated facility location (UFL) that produces precisely such a clustering. The problem however is that [13] uses filtering which involves blowing up the x_{ij} and y_i values, thus violating the matroid-rank packing constraints. Chudak and Shmoys [7] use the same clustering idea but without filtering, using the dual solution to bound the cost. The difficulty here with this approach is that there are terms with negative coefficients in the dual objective function that correspond to the primal matroid-rank constraints. Although [14] showed that it is possible to overcome this difficulty in certain cases, the situation here looks more complicated and it is not clear how to use their techniques.

Instead, we use the clustering technique of Charikar et al. [5] to cluster clients and first obtain a *half-integral solution* $(\hat{x}, \hat{y})$, that is, every $\hat{x}_{ij}, \hat{y}_i \in \left\{0, \frac{1}{2}, 1\right\}$, to the modified instance with cluster centers, losing a factor of 3. Further, any solution here will give a solution to the original instance while increasing the cost by at most $4 \cdot OPT$. Now we use the clustering method of [13] *without any filtering*, since the half-integral solution $(\hat{x}, \hat{y})$ is essentially already filtered; if client j is assigned to i and i' in $\hat{x}$, then $c_{ij}, c_{i'j} \leq 2(c_{ij}\hat{x}_{ij} + c_{i'j}\hat{x}_{i'j})$. This final step causes us to lose an additive factor equal to the cost of $(\hat{x}, \hat{y})$, so overall we get an approximation ratio of $4 + 3 + 3 = 10$. In Section 3.4, we show that by further exploiting the structure of the half-integral solution, we can give a better bound on the cost of the integer solution and thus obtain an 8-approximation.

We now describe each of these steps in detail; omitted proofs appear in the full version. Let $\bar{C}_j = \sum_i c_{ij}x_{ij}$ denote the cost incurred by the LP solution to assign one unit of demand of client j. Given a vector $v \in \mathbb{R}^{\mathcal{F}}$ and a set $S \subseteq \mathcal{F}$, we use $v(S)$ to denote $\sum_{i \in S} v_i$.

3.2 Obtaining a Half-integral Solution $(\hat{x}, \hat{y})$

Step I: Consolidating Demands around Centers. We first consolidate (or cluster) the demand of clients at certain clients, that we call *cluster centers*. We do not modify the fractional solution (x, y) but only modify the demands so that for some clients j, the demand d_j is "moved" to a "nearby" center k. We assume every client has non-zero demand (we can simply get rid of zero-demand clients).

Set $d'_j \leftarrow 0$ for every j. Consider the clients in increasing order of $\bar{C}_j$. For each client j, if there exists a client k such that $d'_k > 0$ and $c_{jk} \leq 4\max(\bar{C}_j, \bar{C}_k) = 4\bar{C}_k$, set $d'_k \leftarrow d'_k + d_j$, otherwise set $d'_j \leftarrow d_j$. Let $D = \{j \in \mathcal{D} : d'_j > 0\}$. Each client in D is a cluster center. Let $OPT' = \sum_i f_i y_i + \sum_{j \in D, i} d'_j c_{ij} x_{ij}$ denote the cost of (x, y) for the modified instance consisting of the cluster centers.

▶ **Lemma 3.1.** *(i) If $j, k \in D$, then $c_{jk} \geq 4\max(\bar{C}_j, \bar{C}_k)$, (ii) $OPT' \leq OPT$, and (iii) any solution (x', y') to the modified instance can be converted to a solution to the original instance incurring an additional cost of at most $4 \cdot OPT$.*

From now on we focus on the modified instance with client set D and modified demands d'_j. At the very end we will use the above lemma to translate an integer solution to the modified instance to an integer solution to the original instance.

Step II: Transforming to a Half-integral Solution. We define the cluster of a client $j \in D$ to be the set F_j of all facilities i such that j is the center in D closest to i, that is, $F_j = \{i : c_{ij} = \min_{k \in D} c_{ik}\}$, with ties broken arbitrarily. Let $F'_j \subseteq F_j = \{i \in F_j : c_{ij} \leq 2\bar{C}_j\}$. Clearly the sets F_j for $j \in D$ are disjoint. By property (i) of Lemma 3.1, we have that F_j contains all the facilities i such that $c_{ij} \leq 2\bar{C}_j$. So $\sum_{i \in F'_j} x_{ij} = \sum_{i: c_{ij} \leq 2\bar{C}_j} x_{ij} \geq \frac{1}{2}$ by Markov's inequality.

To obtain the half-integral solution, we define a suitable vector y' that lies in a polytope with half-integral extreme points and construct a linear function $T(.)$ such that $T(y')$ bounds the cost of a fractional solution. We show that $T(y') \leq 3 \cdot OPT'$. This implies that one can obtain a "better" half-integral vector $\hat{y}$, which we then argue yields a half-integral solution $(\hat{x}, \hat{y})$ to the modified instance of cost at most $T(\hat{y}) \leq T(y')$.

Define $\gamma_j := \min_{i \notin F_j} c_{ij}$, and let $G_j = \{i \in F_j : c_{ij} \leq \gamma_j\}$. Note that $\gamma_j \geq 2\bar{C}_j$, so $F'_j \subseteq G_j$. Set $y'_i = x_{ij} \leq y_i$ if $i \in G_j$, and $y'_i = 0$ otherwise. Clearly, $y'(F_j) = y'(G_j) \leq 1$. Then y' lies in the following polytope

$$\mathcal{P} := \Big\{ v \in \mathbb{R}_+^{\mathcal{F}} : v(S) \leq r(S) \quad \forall S \subseteq \mathcal{F}, \qquad v(F'_j) \geq \tfrac{1}{2}, \quad v(G_j) \leq 1 \quad \forall j \in D \Big\}. \tag{1}$$

We claim that $\mathcal{P}$ has half-integral extreme points. The easiest way to see this is to note that any extreme point of $\mathcal{P}$ is defined by a linearly independent system of tight constraints comprising some $v(S) = r(S)$ equalities corresponding to a laminar set system, and some $v(F'_j) = \frac{1}{2}$ and $v(G_j) = 1$ equalities. The constraint matrix of this system thus corresponds to equations coming from two laminar set systems; such a matrix is known to be totally unimodular, and hence the vector v satisfying this system must be a half-integral solution. (The full version also gives a proof based on the integrality of the intersection of two submodular polyhedra.)

Given $v \in \mathbb{R}_+^{\mathcal{F}}$, define $T(v) = \sum_i f_i v_i + \sum_j d'_j \big(\sum_{i \in G_j} c_{ij} v_i + 3\gamma_j (1 - \sum_{i \in G_j} v_i)\big)$. Since $y' \in \mathcal{P}$, this implies that we can obtain a half-integral solution $\hat{y}$ such that $T(\hat{y}) \leq T(y')$. Observe that there is at least one facility $i \in F'_j$ with $\hat{y}_i > 0$; we call the facility $i \in F'_j$ nearest to j the *primary facility* of j and set $\hat{x}_{ij} = \hat{y}_i$. Note that every every client in D

has a *distinct* primary facility. If $\hat{y}_i < 1$, then let i' be the facility nearest to j other than i such that $\hat{y}_{i'} > 0$; we call i' the *secondary facility* of j, and set $\hat{x}_{i'j} = 1 - \hat{x}_{ij}$. Define $\hat{C}_j = \sum_i c_{ij}\hat{x}_{ij}$ and $S_j = \{i : \hat{x}_{ij} > 0\}$.

▶ **Lemma 3.2.** *The cost of* $(\hat{x}, \hat{y})$ *is at most* $3 \cdot OPT' \leq 3 \cdot OPT$.

3.3 Converting $(\hat{x}, \hat{y})$ to an Integer Solution

Step III: Clustering. We cluster the clients in D as follows: pick $j \in D$ with smallest $\hat{C}_j$. Remove every client $k \in D$ such that $S_j \cap S_k \neq \emptyset$; we call j the *center* of k and denote it by $\mathsf{ctr}(k)$. Recurse on the remaining set of clients until no client in D is left. Let D' be the set of clients picked—these are the *new cluster centers*. Note that $\mathsf{ctr}(j) = j$ for every $j \in D'$.

Step IV: The Matroid Intersection Problem. For convenience, we will say that every client $j \in D$ has both a primary facility $i_1(j)$ and a secondary facility $i_2(j)$ with $\hat{x}_{i_1(j)j} = \hat{x}_{i_2(j)j} = \frac{1}{2}$, with the understanding that if j does not have a secondary facility then $i_1(j) = i_1(j)$, and so $\hat{x}_{i_1(j)j} = 1$. Then we have $\hat{C}_j = \frac{1}{2}\left(c_{i_1(j)j} + c_{i_2(j)j}\right)$ and $c_{i_1(j)j} \leq \hat{C}_j \leq c_{i_2(j)j} \leq 2\hat{C}_j$.

For $i \in \mathcal{F}$, define $\hat{y}'_i = \hat{x}_{ij} \leq \hat{y}_i$ if $i \in S_j$ where $j \in D'$, and $\hat{y}'_i = \hat{y}_i$ otherwise. Then $\hat{y}'$ lies in the polytope

$$\mathcal{R} := \left\{z \in \mathbb{R}_+^{\mathcal{F}} : z(S) \leq r(S) \quad \forall S \subseteq \mathcal{F}, \qquad z(S_j) = 1 \quad \forall j \in D'\right\}. \tag{2}$$

Observe that $\mathcal{R}$ is the intersection of the matroid polytope for M with the matroid base polytope for the partition matroid defined by the S_j sets for $j \in D'$. This polytope is known to have integral extreme points. Similar to Step II, we define a linear function $H(z) = \sum_i f_i z_i + \sum_{k \in D} A_k(z)$, where

$$A_k(z) = \begin{cases} \sum_{i \in S_{\mathsf{ctr}(k)}} d'_k c_{ik} z_i & \text{if } i_1(k) \in S_{\mathsf{ctr}(k)} \\ \sum_{i \in S_{\mathsf{ctr}(k)}} d'_k c_{ik} z_i + d'_k\left(c_{i_1(k)k} - c_{i_2(k)k}\right) z_{i_1(k)} & \text{otherwise.} \end{cases}$$

Since $\mathcal{R}$ is integral, we can find an integer point $\tilde{y} \in \mathcal{R}$ such that $H(\tilde{y}) \leq H(\hat{y}')$. This yields an integer solution $(\tilde{x}, \tilde{y})$ to the instance with client set D, where we assign each client $j \in D'$ to the unique facility opened from S_j, and each client $k \in D \setminus D'$ either to $i_1(k)$ if it is open (i.e., $\tilde{y}_{i_1(k)} = 1$), or to the facility opened from $S_{\mathsf{ctr}(k)}$. In Lemma 3.3 we prove that the cost of this integer solution is at most $H(\tilde{y})$, and in Lemma 3.4 we show that $H(\hat{y}')$ is at most twice the cost of $(\hat{x}, \hat{y})$ and hence, at most $6 \cdot OPT$ (by Lemma 3.2). Combined with Lemma 3.1, this yields Theorem 3.5.

▶ **Lemma 3.3.** *The cost of* $(\tilde{x}, \tilde{y})$ *is at most* $H(\tilde{y}) \leq H(\hat{y}')$.

▶ **Lemma 3.4.** $H(\hat{y}')$ *is at most twice the cost of* $(\hat{x}, \hat{y})$.

▶ **Theorem 3.5.** *The integer solution* $(\tilde{x}, \tilde{y})$ *translates to an integer solution to the original instance of cost at most* $10 \cdot OPT$.

3.4 Improvement to 8-Approximation

The procedure described in Section 3.3 shows that *any* half-integral solution can be rounded to an integral one losing a factor of 2 in the cost. We obtain an improved approximation ratio of 8 by exploiting the structure leading to the half-integral solution obtained in Section 3.2. The key to the improvement comes from the following observation (in various flavors). Consider a non-cluster-center $k \in D' \setminus D$ with $\mathsf{ctr}(k) = j$. Let i be a facility serving both j and k.

Suppose i is not the primary facility of k. Without any further information, we can only say that $c_{jk} \leq c_{ij} + c_{ik} \leq 3\gamma_j + 3\gamma_k$. However, if we define our half-integral solution by setting the secondary facility of k to be the primary facility of the client (in D) nearest to k, then we have the better bound $c_{jk} \leq 2\gamma_j + 2\gamma_k$, which yields an improved bound for k's assignment cost. To push this observation through, we will "couple" the rounding steps used to obtain the half-integral and integral solutions: we tailor the function $T(.)$ (defined in Step II above) so as to allow one to bound the total cost of the final integral solution obtained. Also, we use a different criterion for selecting a cluster center in the clustering performed in Step III.

The first step is the same as Step I in Section 3.2. Recall that the new client-set is D with demands $\{d'_j\}_{j\in D}$, OPT' is the cost of (x, y) for the modified instance, and for each $j \in D$ we define $F_j = \{i : c_{ij} = \min_{k\in D} c_{ik}\}$, $F'_j = \{i \in F_j : c_{ij} \leq 2\bar{C}_j\}$, $\gamma_j = \min_{i\notin F_j} c_{ij}$, and $G_j = \{i \in F_j : c_{ij} \leq \gamma_j\}$.

A1. Obtaining a Half-integral Solution. Set $y'_i = x_{ij} \leq y_i$ if $i \in G_j$, and $y'_i = 0$ otherwise. We define $T(v) = \sum_i f_i v_i + \sum_j d'_j \big(2\sum_{i\in G_j} c_{ij} v_i + 4\gamma_j(1 - \sum_{i\in G_j} v_i)\big)$ for $v \in \mathbb{R}^{\mathcal{F}}_+$ with some hindsight. Since y' lies in the half-integral polytope $\mathcal{P}$ (see (1)), we can obtain a half-integral $\hat{y}$ such that $T(\hat{y}) \leq T(y')$.

For each client $j \in D$, define $\sigma(j) = j$ if $\hat{y}(G_j) = 1$, and $\sigma(j) = \arg\min_{k\in D:k\neq j} c_{jk}$ otherwise (breaking ties arbitrarily). Note that $c_{j\sigma(j)} \leq 2\gamma_j$. As before, we call the facility i nearest to j with $\hat{y}_i > 0$ the primary facility of j and denote it by $i_1(j)$; we set $\hat{x}_{i_1(j)j} = \hat{y}_{i_1(j)}$. Note that $i_1(j) \in F'_j$. If $\hat{y}_{i_1(j)} < 1$ and $\hat{y}(G_j) = 1$, let i' be the fractionally open facility other than $i_1(j)$ nearest to j; otherwise, if $\hat{y}_{i_1(j)} < 1$ and $\hat{y}(G_j) < 1$, (so $\sigma(j) \neq j$ and $\hat{y}_{i_1(j)} = \frac{1}{2}$), let i' be the primary facility of $\sigma(j)$. We call i' the secondary facility of j, and denote it by $i_2(j)$. Again, for convenience, we consider j as having both a primary and secondary facility and $\hat{x}_{i_1(j)j} = \hat{x}_{i_2(j)j} = \frac{1}{2}$, with the understanding that if $\hat{y}_{i_1(j)} = 1$, then $i_2(j) = i_1(j)$ and $\hat{x}_{i_1(j)j} = 1$. Let $S_j = \{i : \hat{x}_{ij} > 0\} = \{i_1(j), i_2(j)\}$.

A2. Clustering and Rounding to an Integral Solution. For each $j \in D$, define $C'_j = \big(c_{i_1(j)j} + c_{j\sigma(j)} + c_{i_2(j)\sigma(j)}\big)/2$. We cluster clients as in Step III in Section 3.3, except that we repeatedly pick the client with smallest C'_j among the remaining clients to be the cluster center. As before, let D' denote the set of cluster centers, and let $\mathsf{ctr}(k) = j \in D'$ for $k \in D$ if k was removed in the clustering process because j was chosen as a cluster center and $S_j \cap S_k \neq \emptyset$.

Similar to Step IV in Section 3.3, for each $i \in \mathcal{F}$, define $\hat{y}'_i = \hat{x}_{ij} \leq \hat{y}_i$ if $i \in S_j$ where $j \in D'$ and $\hat{y}'_i = \hat{y}_i$ otherwise. For $z \in \mathbb{R}^{\mathcal{F}}_+$, define $H(z) = \sum_i f_i z_i + \sum_{k\in D} L_k(z)$, where $L_k(z)$ is $\sum_{i\in S_{\mathsf{ctr}(k)}} d'_k c_{ik} z_i$ if $i_1(k) \in S_{\mathsf{ctr}(k)}$, and $\sum_{i\in S_{\mathsf{ctr}(k)}} d'_k \big(c_{k\sigma(k)} + c_{i\sigma(k)}\big) z_i + d'_k \big(c_{i_1(k)k} - c_{k\sigma(k)} - c_{i_1(\sigma(k))\sigma(k)}\big) z_{i_1(k)}$ otherwise. Since $\hat{y}'$ lies in the integral polytope $\mathcal{R}$ (see (2)), we can obtain an integral vector $\tilde{y}$ such that $H(\tilde{y}) \leq H(\hat{y}')$, and a corresponding integer solution $(\tilde{x}, \tilde{y})$ (as in Step IV in Section 3.3).

Analysis. The 8-approximation guarantee (Theorem 3.8) follows directly by combining Lemmas 3.6 and 3.7 with Lemma 3.1.

▶ **Lemma 3.6.** *We have* $T(\hat{y}) \leq T(y') \leq 4 \cdot OPT' \leq 4 \cdot OPT$.

Proof. We know that $T(\hat{y}) \leq T(y')$ and $OPT' \leq OPT$. We have $OPT' = \sum_i f_i y_i + \sum_j d'_j \bar{C}_j$, and for any $j \in D$, we have $\bar{C}_j = \sum_{i\in G_j} c_{ij} x_{ij} + \sum_{i\notin G_j} c_{ij} x_{ij} \geq \sum_{i\in G_j} c_{ij} x_{ij} + \gamma_j(1 - \sum_{i\in G_j} x_{ij})$ by the definition of γ_j. So $T(y')$ is at most

$$\sum_i f_i y_i + \sum_j d'_j \Big(\sum_{i\in G_j} c_{ij} x_{ij} + 4\gamma_j(1 - \sum_{i\in G_j} x_{ij})\Big) \leq \sum_i f_i y_i + 4\sum_j d'_j \bar{C}_j. \qquad ◀$$

▶ **Lemma 3.7.** *The cost of* $(\tilde{x},\tilde{y})$ *is at most* $H(\tilde{y}) \le H(\hat{y}')$*, and* $H(\hat{y}') \le T(\hat{y})$*.*

Proof. We first argue that the cost of $(\tilde{x},\tilde{y})$ is at most $H(\tilde{y})$. The facility opening cost is $\sum_i f_i\tilde{y}_i$. The assignment cost of a client $j \in D'$ is $\sum_{i\in S_j} d'_j c_{ij}\tilde{y}_i = L_j(\tilde{y})$. Consider a client $k \in D \setminus D'$ with $\mathsf{ctr}(k) = j$. Let $i' = i_1(j)$, $i'' = i_2(j)$. If $\tilde{y}_{i_1(k)} = 0$ or $i_1(k) \in S_j$, then $L_k(\tilde{y})$ is at least $d'_k \sum_{i\in S_j} c_{ik}\tilde{y}_i$. So suppose $\tilde{y}_{i_1(k)} = 1$ and $i_1(k) \notin S_j$. Then the assignment cost of k is $d'_k c_{i_1(k)k}$, and since $c_{i\sigma(k)} \ge c_{i_1(\sigma(k))\sigma(k)}$ for every $i \in S_j$, we have $L_k(\tilde{y}) \ge d'_k c_{i_1(k)k}$.

We now show that $H(\hat{y}') \le T(\hat{y})$. Define $B_j(\hat{y}) := d'_j\big(2\sum_{i\in G_j} c_{ij}\hat{y}_i + 4\gamma_j(1 - \hat{y}(G_j))\big)$. So $T(\hat{y}) = \sum_i f_i\hat{y}_i + \sum_{j\in D} B_j(\hat{y})$. Clearly $\sum_i f_i\hat{y}'_i \le \sum_i f_i\hat{y}_i$. We show that $L_j(\hat{y}') \le B_j(\hat{y})$ for every $j \in D$, which will complete the proof.

We first argue that $d'_jC'_j \le B_j(\hat{y})$ for every $j \in D$. If $\hat{y}(G_j) = 1$, then $d'_jC'_j = \sum_{i\in G_j} d'_j c_{ij}\hat{y}_i \le B_j(\hat{y})$. Otherwise, $\hat{y}(G_j) = \frac{1}{2}$, and $c_{j\sigma(j)} + c_{i_1(\sigma(j))\sigma(j)} \le 3\gamma_j$; so $d'_jC'_j \le d'_j\big(\sum_{i\in G_j} c_{ij}\hat{y}_i + 3\gamma_j(1 - \hat{y}(G_j))\big) \le B_j(\hat{y})$.

For a client $j \in D'$, we have $L_j(\hat{y}') = d'_j\big(c_{i_1(j)j} + c_{i_2(j)j}\big)/2 \le d'_jC'_j \le B_j(\hat{y})$. Now consider a client $k \in D \setminus D'$. Let $j = \mathsf{ctr}(k)$, and $i' = i_1(j)$, $i'' = i_2(j)$. Note that $C'_j \le C'_k$. We consider two cases.

1. $i_1(k) \in S_j$. This means that $i_1(k) = i'' \ne i'$ and $k = \sigma(j)$. So

$$L_k(\hat{y}') = \frac{d'_k}{2}\cdot\big(c_{i''k} + c_{i'k}\big) \le \frac{d'_k}{2}\cdot\big(c_{i'j} + c_{jk} + c_{i''k}\big) = d'_kC'_j \le d'_kC'_k \le B_k(\hat{y}).$$

2. $i_1(k) \notin S_j$. This implies that $\hat{y}(G_k) = \hat{y}_{i_1(k)} = \frac{1}{2}$. Let $\ell = \sigma(k)$ (which is the same as j if $i_2(k) = i_1(j)$). We have $L_k(\hat{y}') = \frac{d'_k}{2}\cdot\big(2c_{k\ell} + c_{i'\ell} + c_{i''\ell} + c_{i_1(k)k} - c_{k\ell} - c_{i_1(\ell)\ell}\big)$. If $\ell = j$, then $L_k(\hat{y}') = \frac{d'_k}{2}\cdot\big(c_{i_1(k)k} + c_{jk} + c_{i''j}\big)$. Notice that $c_{i''j} \le 2C'_j - c_{i'j}$. So we obtain that

$$L_k(\hat{y}') \le \frac{d'_k}{2}\cdot\big(c_{i_1(k)k}+c_{jk}+2C'_j-c_{i'j}\big) \le \frac{d'_k}{2}\cdot\big(c_{i_1(k)k}+c_{jk}+2C'_k-c_{i'j}\big) = d'_k\big(c_{i_1(k)k}+c_{jk}\big).$$

   If $\ell \ne j$, then $i_2(j) = i'' = i_2(k) = i_1(\ell)$, so $\ell = \sigma(j)$, and $c_{i'_j} + c_{j\ell} + c_{i''\ell} = 2C'_j \le 2C'_k = c_{i_1(k)k} + c_{k\ell} + c_{i''\ell}$. So $L_k(\hat{y}') \le \frac{d'_k}{2}\cdot\big(c_{i_1(k)k} + c_{k\ell} + c_{j\ell} + c_{i'j}\big) \le d'_k(c_{i_1(k)k} + c_{k\ell})$. In both cases,

$$L_k(\hat{y}') \le d'_k(c_{i_1(k)k} + c_{k\sigma(k)}\big) \le d'_k\Big(2\sum_{i\in G_k} c_{ik}\hat{y}_i + 4\gamma_k\big(1 - \hat{y}(G_k)\big)\Big) = B_k(\hat{y}).$$ ◀

▶ **Theorem 3.8.** *The integer solution* $(\tilde{x},\tilde{y})$ *translates to an integer solution to the original instance of cost at most* $8\cdot OPT$*.*

▶ Remark. It is easy to modify the above algorithm to obtain a so-called *Lagrangian-multiplier preserving* (LMP) 8-approximation algorithm, that is, where the solution $(\tilde{x},\tilde{y})$ returned satisfies $8\sum_i f_i\tilde{y}_i + \sum_{j\in\mathcal{D},i} d_jc_{ij}\tilde{x}_{ij} \le 8\cdot OPT$. To obtain this, the only change is that we redefine

$$T(v) = 8\sum_i f_iv_i + \sum_j d'_j\big(2\sum_{i\in G_j} c_{ij}v_i + 4\gamma_j(1 - \sum_{i\in G_j} v_i)\big), \qquad H(z) = 8\sum_i f_iz_i + \sum_{k\in D} L_k(z).$$

We now have $T(\hat{y}) \le T(y') \le 8\sum_i f_iy_i + 4\sum_{j\in D} d'_j\bar{C}_j$, and $8\sum_i f_i\tilde{y}_i + \sum_{j\in D,i} d'_jc_{ij}\tilde{x}_{ij} \le H(\tilde{y}) \le H(\hat{y}')$. Also, as before, we have $H(\hat{y}') \le T(\hat{y})$. Thus, we have

$$\begin{aligned} 8\sum_i f_i\tilde{y}_i + \sum_{j\in\mathcal{D},i} d_jc_{ij}\tilde{x}_{ij} &\le 8\sum_i f_i\tilde{y}_i + \sum_{j\in D,i} d'_jc_{ij}\tilde{x}_{ij} + \sum_{j\in\mathcal{D}\setminus D} 4d_j\bar{C}_j \\ &\le 8\sum_i f_iy_i + 4\sum_{j\in D} d_j\bar{C}_j + 8\sum_{j\in\mathcal{D}\setminus D} d_j\bar{C}_j \le 8\cdot OPT. \end{aligned}$$

4 Extensions

4.1 Matroid Median with Two Matroids

A natural extension of matroid median is the *matroid-intersection median* problem, wherein are given two matroids on the facility-set $\mathcal{F}$, and we require the set of open facilities to be an independent set in both matroids. This problem turns out to be inapproximable to within any multiplicative factor in polytime.

▶ **Theorem 4.1.** *It is NP-complete to decide if an instance of matroid-intersection median has a zero-cost solution; this holds even if one of the matroids is a partition matroid. Hence, no multiplicative approximation is achievable in polytime for this problem unless P=NP.*

Proof. The reduction is from the *NP*-complete *directed Hamiltonian path* problem, wherein we are given a directed graph $D = (N, A)$, and two nodes s, t, and we need to determine if there is a simple (directed) $s \rightsquigarrow t$ path spanning all the nodes. The facility-set in the matroid-intersection median problem is the arc-set A, and every node except t is a client. One of the matroids M is the graphic matroid on the undirected version of D, that is, an arc-set is independent if it is acyclic when we ignore the edge directions. The second matroid M_2 is a partition matroid that enforces that every node other than s has at most one incoming arc. All facility-costs are 0. We set $c_{ij} = 0$ if i is an outgoing arc of j, and ∞ otherwise. Notice that this forms a metric since the sets $\{i : c_{ij} = 0\}$ are disjoint for different clients.

It is easy to see that an $s \rightsquigarrow t$ Hamiltonian path translates to a zero-cost solution to the matroid-intersection median problem. Conversely, if we have a zero-cost solution to matroid-intersection median, then it must open $|N| - 1$ facilities, one for each client. Hence, the resulting edges must form a (spanning) arborescence rooted at s, and moreover, every node other than t must have an outgoing arc. Thus, the resulting edges yield an $s \rightsquigarrow t$ Hamiltonian path. ◀

We consider two extensions of matroid median that are essentially special cases of matroid-intersection median and can be used to model some interesting problems (see Section 5). The techniques developed in Section 3 readily extend and yield an 8-approximation algorithm (in fact, an LMP 8-approximation) for both problems. These extensions may be viewed in some sense as the most-general special cases of matroid-intersection median that one can hope to approximately solve in polytime.

The setup in both extensions is similar. We have a matroid $M = (\mathcal{F}, \mathcal{I})$ on the facility-set (and clients with demands and assignment costs). $\mathcal{F}$ is partitioned into $\mathcal{F}_1 \cup \mathcal{F}_2$ and clients may only be assigned to facilities in $\mathcal{F}_1$; this can be encoded by setting $c_{ij} = \infty$ for all $i \in \mathcal{F}_2$ and $j \in \mathcal{D}$. We also have lower and upper bounds $(\mathit{lb1}, \mathit{ub1})$, $(\mathit{lb2}, \mathit{ub2})$, and $(\mathit{lb}, \mathit{ub})$ on the number of facilities that may be opened from $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}$ respectively. We need to open a feasible set of facilities and assign every client to an open facility so as to minimize the total facility-opening and client-assignment cost. A set $F \subseteq \mathcal{F}$ of facilities is said to be feasible if: (i) $F \in \mathcal{I}$; (ii) $\mathit{lb1} \leq |F \cap \mathcal{F}_1| \leq \mathit{ub1}$, $\mathit{lb2} \leq |F \cap \mathcal{F}_2| \leq \mathit{ub2}$, $\mathit{lb} \leq |F| \leq \mathit{ub}$; and (iii) $F \cap \mathcal{F}_2$ satisfies problem-specific constraints. While the role of $\mathcal{F}_2$ may seem unclear, notice that a non-trivial lower bound on the number of $\mathcal{F}_2$-facilities imposes restrictions on the facilities that may be opened from $\mathcal{F}_1$ due to the matroid M (see, e.g., k-median forest in Section 5).

Two-matroid Median (2MMed). In addition to the above setup, we have another matroid $M_2 = (\mathcal{F}_2, \mathcal{I}_2)$ on $\mathcal{F}_2$ with rank function r_2. A set F of facilities is feasible if it satisfies (i) and (ii) above, and (iii) $F \cap \mathcal{F}_2 \in \mathcal{I}_2$. We may modify the matroids M and M_2 to incorporate the upper bounds *ub* and *ub2* respectively in their definition; *we assume that this has been done in the sequel.* The LP-relaxation for 2MMed is quite similar to (P). We augment (P) with the constraints:

$$y(S) \leq r_2(S) \quad \forall S \subseteq \mathcal{F}_2, \qquad \mathit{lb1} \leq y(\mathcal{F}_1) \leq \mathit{ub1}, \quad \mathit{lb2} \leq y(\mathcal{F}_2), \quad \mathit{lb} \leq y(\mathcal{F}).$$

Let (x, y) denote an optimal solution to this LP, and OPT denote its cost. The rounding procedure dovetails the one in Section 3. The first step is again Step I in Section 3.2. Let D be the new client-set with demands $\{d'_j\}_{j \in D}$, OPT' be the new cost of (x, y), and for each $j \in D$, we define F_j, F'_j, γ_j, and G_j as before. Note that $F_j \subseteq \mathcal{F}_1$ for all $j \in D$. A slight technicality arises in mimicking Step A1 in Section 3.4: setting $y'_i = x_{ij}$ for some facility $i \in G_j$ need not satisfy the lower-bound constraints. We deal with this by "cloning" facilities suitably to obtain: (i) a new $\mathcal{F}'_1$-set $\mathcal{F}_1$, corresponding facility-set $\mathcal{F}' = \mathcal{F}'_1 \cup \mathcal{F}_2$ and facility-opening vector $y \in \mathbb{R}_+^{\mathcal{F}'}$; (ii) a new set $G'_j \subseteq \mathcal{F}'_1$ for all $j \in D$; (iii) a new rank function $r' : 2^{\mathcal{F}'} \mapsto \mathbb{Z}_+$.

We continue with steps A1, A2 in Section 3.4, replacing G_j with G'_j, and using suitable polytopes in place of $\mathcal{P}$ and $\mathcal{R}$ to obtain the half-integral and integral solutions. To obtain a half-integral solution, we define

$$\mathcal{P}' := \Big\{ v \in \mathbb{R}_+^{\mathcal{F}'} : \ v(S) \leq r'(S) \ \ \forall S \subseteq \mathcal{F}', \quad v(S) \leq r_2(S) \ \ \forall S \subseteq \mathcal{F}_2, \quad \mathit{lb} \leq v(\mathcal{F}') \\ \mathit{lb1} \leq v(\mathcal{F}'_1) \leq \mathit{ub1}, \quad \mathit{lb2} \leq v(\mathcal{F}_2), \quad v(F'_j) \geq \tfrac{1}{2}, \ v(G'_j) \leq 1 \quad \forall j \in D \Big\} \tag{3}$$

which contains y. The key observation is that an extreme point of $\mathcal{P}'$ is again defined by a linearly independent system of tight constraints coming from two laminar systems: one consisting of some tight $v(S) \leq r'(S)$ and $\mathit{lb} \leq v(\mathcal{F}') \leq \mathit{ub}$ constraints; the other consisting of some tight $v(S) \leq r_2(S)$ and $\mathit{lb1} \leq v(\mathcal{F}'_1) \leq \mathit{ub1}$, $\mathit{lb2} \leq v(\mathcal{F}_2) \leq \mathit{ub2}$ constraints, and some tight $v(F'_j) \leq \frac{1}{2}$ and $v(G'_j) \geq 1$ constraints. Thus, $\mathcal{P}'$ has half-integral extreme points, and so we can find a half-integral $\hat{y}$ such that $T(\hat{y}) \leq T(y)$, and a corresponding solution $(\hat{x}, \hat{y})$. We round this to an integral solution as in step A2, using the polytope

$$\mathcal{R}' := \Big\{ z \in \mathbb{R}_+^{\mathcal{F}'} : \ z(S) \leq r'(S) \quad \forall S \subseteq \mathcal{F}', \qquad z(S) \leq r_2(S) \quad \forall S \subseteq \mathcal{F}_2 \\ \mathit{lb1} \leq z(\mathcal{F}'_1) \leq \mathit{ub1}, \quad \mathit{lb2} \leq z(\mathcal{F}_2), \quad \mathit{lb} \leq z(\mathcal{F}'), \quad z(S_j) = 1 \ \ \forall j \in D' \Big\} \tag{4}$$

which has integral extreme points. A useful observation is that if $j \in D'$ then we may assume that $\hat{x}_{ij} = \hat{y}_i$ for all $i \in S_j$, and so $\hat{y} \in \mathcal{R}'$. So we obtain an integral vector $\tilde{y}$ such that $H(\tilde{y}) \leq H(\hat{y})$, and hence an integer solution $(\tilde{x}, \tilde{y})$. (Here $T(.)$ and $H(.)$ are as defined in in Section 3.4.) Mimicking Lemmas 3.6 and 3.7, we obtain that $T(\hat{y}) \leq T(y) \leq 4 \cdot OPT'$, and the cost of $(\tilde{x}, \tilde{y})$ is at most $H(\tilde{y}) \leq H(\hat{y}) \leq T(\hat{y})$. Thus, we obtain the following theorem.

▶ **Theorem 4.2.** *The integer solution $(\tilde{x}, \tilde{y})$ yields an integer solution to 2MMed of cost at most $8 \cdot OPT$.*

Laminarity-constrained Matroid Median (LCMMed). In LCMMed, in addition to the common setup, we have a laminar family $\mathcal{L}$ on $\mathcal{F}_2$ and bounds $0 \leq \ell_S \leq u_S$ for every set $S \in \mathcal{L}$; a set F of facilities is feasible if it satisfies (i) and (ii) above, and (iii) $\ell_S \leq |F \cap S| \leq u_S$ for all $S \in \mathcal{L}$, The approach used for 2MMed also works for LCMMed. The only (obvious) changes are that the LP-relaxation, as well as the definition of the polytopes $\mathcal{P}'$ and $\mathcal{R}'$ (in (3) and (4)) now include the laminarity constraints in place of the rank constraints for the second matroid. All other steps and arguments proceed *identically*, and so we obtain an 8-approximation algorithm for laminarity-constrained matroid median.

4.2 Matroid Median with Penalties

This is the generalization of matroid median where are allowed to leave some clients unassigned at the expense of incurring a penalty $d_j\pi_j$ for each unassigned client j. This changes the LP-relaxation (P) as follows. We use a variable z_j for each client $j \in \mathcal{D}$ to denote if we incur the penalty for client j, and modify the assignment constraint for client j to $\sum_i x_{ij} + z_j \geq 1$; also the objective is now to minimize $\sum_i f_i y_i + \sum_j d_j\bigl(\sum_i c_{ij}x_{ij} + \pi_j z_j\bigr)$. Let (x, y, z) denote an optimal solution to this LP and OPT be its value. Krishnaswamy et al. [11] showed that (x, y, z) can be rounded to an integer solution losing a factor of 360. We show that our rounding approach for matroid median can be adapted to yield a substantially improved 24-approximation algorithm. The rounding procedure is similar to the one described in Section 3 for matroid median, except that we now need to deal with the complication that a client need be assigned fractionally to an extent of 1. We defer the algorithm description and its analysis to the full version.

5 Applications

We now show that the various facility location problems listed below can be cast as special cases of matroid median or the extensions considered in Section 4.1. Thus, our 8-approximation algorithms for matroid median and these extensions immediately yield *improved approximation guarantees for all these problems.*

Problem	**Previous best approximation factor**
Data placement problem [2, 3]	10 [3]
Mobile facility location [9, 1] (with general movement costs)	—; *our reduction* and results of [11, 6] yield factors of 16 and 9 $((3+\epsilon)$ [1] for proportional movement costs)
k-median forest [10] (with non-uniform metrics)	16 [10] $((3+\epsilon)$ [10] for related metrics)
Metric-uniform minimum-latency UFL (MLUFL) [4]	10.773 [4]

The Data Placement Problem. We have a set of caches $\mathcal{F}$, a set of data objects $\mathcal{O}$, and a set of clients $\mathcal{D}$. Each cache $i \in \mathcal{F}$ has a capacity u_i. Each client $j \in \mathcal{D}$ has demand d_j for a specific data object $o(j) \in \mathcal{O}$ and has to be assigned to a cache that stores $o(j)$. Storing an object o in cache i incurs a storage cost of f_i^o, and assigning client j to cache i incurs an access cost of $d_j c_{ij}$, where the c_{ij}s form a metric. We want to determine a set of objects $\mathcal{O}(i) \subseteq \mathcal{O}$ to place in each cache $i \in \mathcal{F}$ satisfying $|\mathcal{O}(i)| \leq u_i$, and assign each client j to a cache $i(j)$ that stores object $o(j)$, (i.e., $o(j) \in \mathcal{O}(i(j))$) so as to minimize $\sum_{i\in\mathcal{F}}\sum_{o\in\mathcal{O}(i)} f_i^o + \sum_{j\in\mathcal{D}} d_j c_{i(j)j}$.

Reduction to matroid median. The facility-set in the matroid-median instance is $\mathcal{F} \times \mathcal{O}$. Facility (i, o) denotes that we store object o in cache i, and has cost f_i^o. The client set is $\mathcal{D}$. We set the distance $c_{(i,o)j}$ to be c_{ij} if $o(j) = o$ and ∞ otherwise, thus enforcing that each client j is only assigned to a facility containing object $o(j)$. The new distances form a metric if the c_{ij}s form a metric. The cache-capacity constraints are incorporated via the matroid where a set $S \subseteq \mathcal{F} \times \mathcal{O}$ is independent if $|\{(i', o) \in S : i' = i\}| \leq u_i$ for every $i \in \mathcal{F}$.

Mobile facility location. In the version with general movement costs, the input is a metric space $\bigl(V, \{c_{ij}\}\bigr)$. We have a set $\mathcal{D} \subseteq V$ of clients, with each client j having demand d_j, and a set $\mathcal{F} \subseteq V$ of initial facility locations. A solution moves each facility $i \in \mathcal{F}$ to a final location $s_i \in V$ incurring a movement cost of $w_{is_i} \geq 0$, and assigns each client j to the final location

s of some facility incurring an assignment cost of $d_j c_{sj}$. The goal is to minimize the sum of all the movement and assignment costs.

Reduction to matroid median. We define the facility-set in the matroid-median instance to be $\mathcal{F} \times V$. Facility (i, s_i) denotes that $i \in \mathcal{F}$ is moved to location $s_i \in V$, and has cost w_{is_i} (note that s_i could be i). The client-set is unchanged, and we set $c_{(i,s_i)j}$ to be $c_{s_i j}$ for every facility $(i, s_i) \in \mathcal{F} \times V$ and client $j \in \mathcal{D}$. These new distances form a metric: we have $c_{(i,s_i)j} \le c_{(i,s_i)k} + c_{(i',s_{i'})k} + c_{(i',s_{i'})j}$ since $c_{s_i j} \le c_{s_i k} + c_{s_{i'} k} + c_{s_{i'} j}$. The constraint that a facility in $\mathcal{F}$ can only be moved to one final location can be encoded by defining a matroid where a set $S \subseteq \mathcal{F} \times V$ is said to be independent if $|\{(i', s) \in S : i' = i\}| \le 1$ for all $i \in \mathcal{F}$.

k-median Forest. In the non-uniform version, we have two metric spaces $\big(V, \{c_{uv}\}\big)$ and $\big(V, \{d_{uv}\}\big)$. The goal is to find $S \subseteq V$ with $|S| \le k$ and assign every node $j \in V$ to $i(j) \in S$ so as to minimize $\sum_j c_{i(j)j} + d\big(\mathsf{MST}(V/S)\big)$, where $\mathsf{MST}(V/S)$ is a minimum spanning forest where each component contains a node of S.

Reduction to 2MMed (or LCMMed). We actually reduce a generalization, where there is an "opening cost" $f_i \ge 0$ incurred for including i in S; the resulting instance is also an LCMMed instance. We add a root r to V. The facility-set $\mathcal{F}$ is the edge-set of the complete graph on $V \cup \{r\}$. The client-set is $\mathcal{D} := V$. Selecting a facility (r, i) denotes that $i \in S$, and selecting a facility (u, v), where $u, v \ne r$, denotes that (u, v) is part of $\mathsf{MST}(V/S)$. We let $\mathcal{F}_1$ be the edges incident to r, and $\mathcal{F}_2$ be the remaining edges. The cost of a facility $(r, i) \in \mathcal{F}_1$ is f_i; the cost of a facility $(u, v) \in \mathcal{F}_2$ is d_{uv}. The client-facility distances are given by $c_{(r,i)j} = c_{ij}$ and $c_{ej} = \infty$ for every $e \in \mathcal{F}_2$. Note that these $\{c_{ej}\}$ distances form a metric. We let M be the graphic matroid of the complete graph on $V \cup \{r\}$. We impose a lower bound of $|V|$ on the number of facilities opened from $\mathcal{F}$, and an upper bound of k on the number of facilities opened from $\mathcal{F}_1$. The matroid M_2 on $\mathcal{F}_2$ is the vacuous one where every set is independent.

A feasible solution to the 2MMed instance corresponds to a spanning tree on $V \cup \{r\}$ where r has degree at most k. This yields a solution to k-median forest of no-greater cost, where the set S is the set of nodes adjacent to r in this edge-set. Conversely, it is easy to see that a solution S to the k-median forest instance yields a 2MMed solution of no-greater cost.

Metric Uniform MLUFL. We have a set $\mathcal{F}$ of facilities with opening costs $\{f_i\}_{i\in\mathcal{F}}$, and a set $\mathcal{D}$ of clients with assignment costs $\{c_{ij}\}_{j\in\mathcal{D}, i\in\mathcal{F}}$, where the c_{ij}s form a metric. Also, we have a monotone latency-cost function $\lambda : \mathbb{Z}_+ \mapsto \mathbb{R}_+$. The goal is to choose a set $F \subseteq \mathcal{F}$ of facilities to open, assign each open facility $i \in F$ a distinct time-index $t_i \in \{1, \ldots, |\mathcal{F}|\}$, and assign each client j to an open facility $i(j) \in F$ so as to minimize $\sum_{i\in F} f_i + \sum_{j\in\mathcal{D}}\big(c_{i(j)j} + \lambda(t_{i(j)})\big)$.

Reduction to Matroid Median. We define the facility-set to be $\mathcal{F} \times \{1, \ldots, |\mathcal{F}|\}$ and the matroid on this set to encode that a set S is independent if $|\{(i, t') \in S : t' = t\}| \le 1$ for all $t \in \{1, \ldots, |\mathcal{F}|\}$. We set $f_{(i,t)} = f_i$ and $c_{(i,t),j} = c_{ij} + \lambda(t)$; note that these distances form a metric. It is easy to see that we can convert any matroid-median solution to one where we open at most one (i, t) facility for any given i without increasing the cost, and hence, the matroid-median instance correctly encodes metric uniform MLUFL.

6 Knapsack Median

We now consider the *knapsack median problem* [11, 12], wherein instead of a matroid on the facility-set, we have a knapsack constraint on the facility-set. Kumar [12] obtained the first constant-factor approximation algorithm for this problem, and [6] obtained an improved

34-approximation algorithm. We consider a somewhat more-general version of knapsack median, wherein each facility i has a facility-opening cost f_i and a *weight* w_i, and we have a knapsack constraint $\sum_{i \in F} w_i \leq B$ constraining the total weight of open facilities. We leverage the ideas from our improved rounding procedure for matroid median to obtain an improved 32-approximation algorithm for this (generalized) knapsack-median problem.

We may assume that we know the maximum facility-opening cost f^{opt} of a facility opened by an optimal solution, so in the sequel we assume that $f_i \leq f^{opt}$, $w_i \leq B$ for all facilities $i \in \mathcal{F}$. Krishnaswamy et al. [11] showed that the natural LP-relaxation for knapsack median has a bad integrality gap; this holds even after augmenting the natural LP with knapsack-cover inequalities. To circumvent this difficulty, Kumar [12] proposed the following lower bound, which we also use. Suppose that we have an estimate C^{opt} within a $(1+\epsilon)$-factor of the connection cost of an optimal solution (which we can obtain by enumerating all powers of $(1+\epsilon)$). Then, defining $U_j := \arg\max\{z : \sum_k d_k \max\{0, z - c_{jk}\} \leq C^{opt}\}$, Kumar argued that the constraint $x_{ij} = 0$ if $c_{ij} > U_j$ is valid for the knapsack median instance. We augment the natural LP-relaxation with these constraints to obtain the following LP (K-P).

$$\begin{aligned}
\min \quad & \sum_i f_i y_i + \sum_j \sum_i d_j c_{ij} x_{ij} && && \text{(K-P)} \\
\text{s.t.} \quad & \sum_i x_{ij} = 1 && \forall j \\
& x_{ij} \leq y_i && \forall i, j \\
& \sum_i w_i y_i \leq B \\
& x_{ij}, y_i \geq 0 && \forall i, j; \qquad x_{ij} = 0 \quad \text{if } c_{ij} > U_j.
\end{aligned}$$

Let (x, y) be an optimal solution to (K-P) and OPT be its value. Let $\bar{C}_j = \sum_i c_{ij} x_{ij}$. Note that if our estimate C^{opt} is correct, then OPT is at most the optimal value opt for the knapsack median instance. We show that (x, y) can be rounded to an integer solution of cost $f^{opt} + 4C^{opt} + 28 \cdot OPT$. Thus, if consider all possible choices for C^{opt} in powers of $(1+\epsilon)$ and pick the solution returned with least cost, we obtain a solution of cost at most $(32+\epsilon)$ times the optimum. The rounding procedure first obtains a nearly half-integral solution whose cost is within a constant-factor of the optimum, which then turns out to be easy to round to an integral solution. The resulting algorithm and analysis is simpler than that in [12, 6]. A detailed description of our algorithm is as follows.

K1. Consolidating Demands. We start by consolidating demands as in Step I in Section 3.2. We now work with the client set D and the demands $\{d'_j\}_{j \in D}$. For $j \in D$, we use $M_j \subseteq \mathcal{D}$ to denote the set of clients (including j) whose demands were moved to j. Note that the M_js partition $\mathcal{D}$. Let OPT' denote the cost of (x, y) for this modified instance. As before, for each $j \in D$ we define $F_j = \{i : c_{ij} = \min_{k \in D} c_{ik}\}$, $F'_j = \{i \in F_j : c_{ij} \leq 2\bar{C}_j\}$, $\gamma_j = \min_{i \notin F_j} c_{ij}$, and $G_j = \{i \in F_j : c_{ij} \leq \gamma_j\}$.

K2. Obtaining a Nearly Half-integral Solution. Set $y'_i = x_{ij} \leq y_i$ if $i \in G_j$, and $y'_i = 0$ otherwise. Let $\mathcal{F}' = \bigcup_{j \in D} G_j$. In the sequel, we will only consider facilities in $\mathcal{F}'$. Consider the following polytope:

$$\mathcal{K} := \Big\{ v \in \mathbb{R}_+^{\mathcal{F}'} : v(F'_j) \geq \tfrac{1}{2}, \quad v(G_j) \leq 1 \quad \forall j \in D, \qquad \sum_i w_i v_i \leq B \Big\}. \tag{5}$$

Define $K(v) = \sum_i 2 f_i v_i + \sum_j d'_j \big(2 \sum_{i \in G_j} c_{ij} v_i + 8\gamma_j (1 - v(G_j))\big)$ for $v \in \mathbb{R}_+^{\mathcal{F}'}$. Since $y' \in \mathcal{K}$, we can efficiently obtain an extreme point $\hat{y}$ of $\mathcal{K}$ such that $K(\hat{y}) \leq K(y')$, the support of $\hat{y}$ is a subset of the support of y', and all constraints that are tight under y' remain tight under $\hat{y}$. Thus, if $i \in G_j$ and $\hat{y}_i > 0$, then $y'_i > 0$ and so $c_{ij} \leq U_j$. Also, if $\hat{y}(G_j) < 1$ then $y'(G_j) < 1$, and so $\gamma_j \leq U_j$. We show in Lemma 6.1 that there is *at most one* client, which we call the *special client* and denote by s, such that G_s contains a facility i with $\hat{y}_i \notin \{0, \frac{1}{2}, 1\}$.

As in Section 3.4, for each client $j \in D$, define $\sigma(j) = j$ if $\hat{y}(G_j) = 1$, and $\sigma(j) = \arg\min_{k \in D: k \neq j} c_{jk}$ otherwise (breaking ties arbitrarily). Note that $c_{j\sigma(j)} \leq 2\gamma_j$. We now define the primary and secondary facilities of each client $j \in D$, which we denote by $i_1(j)$ and $i_2(j)$ respectively. If j is not the special client s, then $i_1(j)$ is the facility i nearest to j with $\hat{y}_i > 0$; otherwise, $i_1(j) = \arg\min_{i \in F'_j: \hat{y}_i > 0} w_i$ (breaking ties arbitrarily). If $\hat{y}_{i_1(j)} = 1$, then we set $i_2(j) = i_1(j)$. If $\hat{y}(G_j) < 1$, we set $i_2(j) = i_1(\sigma(j))$. If $\hat{y}_{i_1(j)} < \hat{y}(G_j) = 1$, we set $i_2(j)$ to: the half-integral facility in G_j other than $i_1(j)$ that is nearest to j if $j \neq s$; and the facility with smallest weight among the facilities $i \in G_j$ with $\hat{y}_i > 0$ (which could be the same as $i_1(j)$) if $j = s$. Define $S_j = \{i_1(j), i_2(j)\}$.

To gain some intuition, observe that the facilities $i_1(j)$ and $i_2(j)$ naturally yield a half-integral solution, where these facilities are open to an extent of $\frac{1}{2}$ and j is assigned to them to an extent of $\frac{1}{2}$; as before, if $i_1(j) = i_2(j)$, then this means that $i_1(j)$ is open to an extent of 1 and j is assigned completely to $i_1(j)$. The choice of the primary and secondary facilities ensures that this solution is feasible. (We do not however modify $\hat{y}$ as indicated above.)

K3. Clustering and Rounding to an Integral Solution. This step is quite straightforward. We define C'_j for $j \in D$, and cluster clients in D exactly as in step A2 in Section 3.4, and we open the facility with smallest weight within each cluster. Finally, we assign each client to the nearest open facility. Let $(\tilde{x}, \tilde{y})$ denote the resulting solution. Recall that D' is the set of cluster centers, and for $k \in D$, $\mathsf{ctr}(k)$ denotes the client in D due to which k was removed in the clustering process (so $\mathsf{ctr}(j) = j$ for $j \in D'$).

Analysis. We call a facility i half-integral (with respect to the vector $\hat{y}$ obtained in step K2) if $\hat{y}_i \in \{0, \frac{1}{2}, 1\}$ and fractional otherwise.

▶ **Lemma 6.1.** *The extreme point $\hat{y}$ of $\mathcal{K}$ obtained in step K2 is such that there is at most one client, called the* special client *and denoted by s, such that G_s contains fractional facilities. Moreover, if $\frac{1}{2} < \hat{y}(G_s) < 1$, then there is one exactly one facility $i \in F'_s$ such that $\hat{y}_i > 0$.*

Proof. Since $\hat{y}$ is an extreme point, it is well known that the submatrix A' of the constraint matrix whose columns correspond to the non-zero $\hat{y}_i$s and rows correspond to the tight constraints under $\hat{y}$ has full column-rank. The rows and columns of A' may be accounted for as follows. Each client $j \in D$ contributes: (i) a non-empty disjoint set of columns corresponding to the positive $\hat{y}_i$s in G_j; and (ii) a possibly-empty disjoint set of at most two rows corresponding to the tight constraints $\hat{y}(F'_j) = \frac{1}{2}$ and $\hat{y}(G_j) = 1$. This accounts for all columns of A'. There is at most one remaining row of A', which corresponds to the tight constraint $\sum_i w_i \hat{y}_i = B$.

Let p_j and q_j denote respectively the number of columns and rows contributed by $j \in D$. First, note that $p_j \geq q_j$ for all $j \in D$. This is clearly true if $q_j \leq 1$; if $q_j = 2$, then $\hat{y}(F'_j) = \frac{1}{2}$, $\hat{y}(G_j) = 1$, so both F'_j and G_j must have at least one positive $\hat{y}_i$. Also, note that if $p_j = q_j$, then G_j contains only half-integral facilities. Since $\sum_j p_j \leq \sum_j q_j + 1$, there can be at most one client such that $p_j > q_j$; we let this be our special client s. Note that we must have $p_s = q_s + 1$.

If $\frac{1}{2} < \hat{y}(G_s) < 1$ then: (i) $q_s = 0$, so $p_s = 1$; or (ii) $q_s = 1$, so $p_s = 2$, and since $\hat{y}(F'_s) = \frac{1}{2} < \hat{y}(G_s)$, both F'_s and G_s contain exactly one positive $\hat{y}_i$. ◀

It is easy to adapt the proof of Lemma 3.6, and obtain that $K(\hat{y}) \leq K(y') \leq 8 \cdot OPT' \leq 8 \cdot OPT$. Next, we prove our main result: the integer solution $(\tilde{x}, \tilde{y})$ computed is feasible and its cost for the modified instance is at most $K(\hat{y}) + f^{opt} + 4C^{opt} + 16 \cdot OPT$. Thus, "moving" the consolidated demands back to their original locations yields a solution of cost at most $(32 + \epsilon) \cdot opt$ for the correct guess of f^{opt} and C^{opt}. The following claims will be useful.

▶ **Claim 6.2.** *If $\hat{y}(G_j) = 1$ for some $j \in D$, then (we may assume that) j is a cluster center.*

Proof. Let $i' = i_1(j)$, $i'' = i_2(j)$. Let $k \in D$ be such that $S_k \cap S_j \neq \emptyset$. Then $\sigma(k) = j$. So $2(C'_k - C'_j) = c_{i_1(k)k} + c_{jk} - c_{i_2(j)j} \geq c_{i_1(k)j} - c_{i_2(j)j} \geq 0$ since $i_2(k) \notin G_j$. ◀

▶ **Claim 6.3.** *For any client $j \in D$, we have $d'_j U_j \leq C^{opt} + 4 \cdot OPT$.*

Proof. By definition, $\sum_k d_k \max\{0, U_j - c_{jk}\} \leq C^{opt}$. So $d'_j U_j = \sum_{k \in M_j} d_k U_j$, which equals

$$\sum_{k \in M_j} d_k(U_j - c_{jk}) + \sum_{k \in M_j} d_k c_{jk} \leq C^{opt} + \sum_{k \in M_j} 4 d_k \bar{C}_k \leq C^{opt} + 4 \cdot OPT. \qquad ◀$$

▶ **Theorem 6.4.** *The solution $(\tilde{x}, \tilde{y})$ computed in step K3 for the modified instance is feasible and has cost at most $K(\hat{y}) + f^{opt} + 4C^{opt} + 16 \cdot OPT$.*

Proof. Let $B_j(v) = d'_j\big(2\sum_{i \in G_j} c_{ij} v_i + 8\gamma_j(1 - v(G_j)\big)$ for $v \in \mathbb{R}_+^{\mathcal{F}'}$. So $K(\hat{y}) = 2\sum_i f_i \hat{y}_i + \sum_j B_j(\hat{y})$. Recall that $S_j = \{i_1(j), i_2(j)\}$ for every $j \in D$.

We first prove feasibility and bound the total facility-opening cost. Consider a cluster centered at j. Let $i' = i_1(j)$, $i'' = i_2(j)$. Let $\hat{i}$ be the facility opened from S_j. If $\hat{y}(S_j) = 1$, then $w_{\hat{i}} \leq \sum_{i \in S_j} w_i \hat{y}_i$. Otherwise, either $j = s$ or $\sigma(j) = s$. If $j = \sigma(j) = s$, then $\hat{i}$ is the least-weight facility in G_j. Otherwise, if $j = s$ then $\hat{i}$ is the least-weight facility in $F'_j \cup \{i_2(j)\}$ and $\hat{y}(F'_j) + \hat{y}_{i_2(j)} \geq 1$; finally, if $j \neq \sigma(j) = s$ then $\hat{i}$ is the least-weight facility in $\{i_1(j)\} \cup F'_{\sigma(j)}$ and $\hat{y}_{i_1(j)} + \hat{y}(F'_{\sigma(j)}) \geq 1$. Since $S_j \subseteq G_j \cup G_{\sigma(j)}$, in every case, we have $w_{\hat{i}} \leq \sum_{i \in G_j \cup G_{\sigma(j)}} w_i \hat{y}_i$.

If all facilities in S_j are half-integral, then $f_{\hat{i}} \leq 2\sum_{i \in S_j} f_i \hat{y}_i \leq 2\sum_{i \in G_j \cup G_{\sigma(j)}} f_i \hat{y}_i$. Otherwise, we have $j = s$ or $\sigma(j) = s$, and we bound $f_{\hat{i}}$ by f^{opt}.

Note that if $k \in D'$ is some other cluster center, then $G_j \cup G_{\sigma(j)}$ is disjoint from $G_k \cup G_{\sigma(k)}$. If not, then we must have $\sigma(j) = k$ or $\sigma(k) = j$ or $\sigma(j) = \sigma(k)$, which yields the contradiction that $S_j \cap S_k \neq \emptyset$. So summing over all clusters, we obtain that the total weight of open facilities is at most $\sum_{j \in D'} \sum_{i \in G_j \cup G_{\sigma(j)}} w_i \hat{y}_i \leq \sum_i w_i \hat{y}_i \leq B$, and the facility opening cost is at most $2\sum_i f_i \hat{y}_i + f^{opt}$.

We now bound the total client-assignment cost. Fix a client $j \in D'$. The assignment cost of j is at most $d'_j c_{i_2(j)j}$. Note that $c_{i_2(j)j} \leq 3U_j$. If $j \neq s$, then $B_j(\hat{y}) \geq d'_j c_{i_2(j)j}$: this holds if $\hat{y}(G_j) = 1$ since $\hat{y}_{i_2(j)} \geq \frac{1}{2}$; otherwise, $B_j(\hat{y}) \geq 4d'_j \gamma_j \geq d'_j c_{i_2(j)j}$. If $j = s$, then its assignment cost is at most $3d'_j U_j \leq 3C^{opt} + 12 \cdot OPT$ (Claim 6.3).

Now consider $k \in D \setminus D'$. Let $j = \mathsf{ctr}(k)$, and $i' = i_1(j)$, $i'' = i_2(j)$. We consider two cases.

1. $i_1(k) \in S_j$. Then $k = \sigma(j)$ and k's assignment cost is at most $d'_k c_{i_2(k)k}$. As above, this is bounded by $B_k(\hat{y})$ if $k \neq s$, and by $3C^{opt} + 12 \cdot OPT$ otherwise.
2. $i_1(k) \notin S_j$. Let $\ell = \sigma(k)$. We claim that the assignment cost of k is at most $d'_k\big(c_{i_1(k)k} + 4\gamma_k\big)$. To see this, first suppose $\ell \neq j$, and so $\ell = \sigma(j)$. Then, k's assignment cost is at most $d'_k\big(c_{k\ell} + c_{\ell j} + c_{i'j}\big) \leq d'_k\big(2c_{k\ell} + c_{i_1(k)k}\big) \leq d'_k\big(c_{i_1(k)k} + 4\gamma_k\big)$, where the first inequality follows since $C'_j \leq C'_k$. If $\ell = j$, then $i_2(k) = i_1(j) = i'$ and k's assignment cost is at most $d'_k\big(c_{jk} + c_{j\sigma(j)} + c_{i_2(j)\sigma(j)}\big) \leq d'_k\big(c_{i_1(k)} + 2c_{jk}\big) \leq d'_k\big(c_{i_1(k)k} + 4\gamma_k\big)$, where the first inequality again follows from $C'_j \leq C'_k$.

Since $k \notin D'$, we have $\hat{y}(G_k) < 1$ (by Claim 6.2). So $y'(G_k) < 1$ and $\gamma_k \leq U_k$. If $k \neq s$, then $B_k(\hat{y}) \geq d'_k\bigl(c_{i_1(k)k} + 4\gamma_k\bigr)$. If $k = s$ and $\hat{y}(G_k) = \frac{1}{2}$, then $B_k(\hat{y}) \geq 4d'_k\gamma_k$ and $d'_k c_{i_1(k)k} \leq d'_k U_k$. Otherwise, by Lemma 6.1, we have $\hat{y}_{i_1(k)} > \frac{1}{2}$, and so $B_k(\hat{y}) \geq d'_k c_{i_1(k)k}$ and $4d'_k\gamma_k \leq 4d'_k U_k$. Taking all cases into account, we can bound k's assignment cost by $B_k(\hat{y})$ if $k \neq s$, and by $B_k(\hat{y}) + 4d'_k U_k \leq B_k(\hat{y}) + 4C^{opt} + 16 \cdot OPT$ if $k = s$.

Putting everything together, the total cost of $(\tilde{x}, \tilde{y})$ is at most $2\sum_i f_i\hat{y}_i + \sum_j B_j(\hat{y}) + f^{opt} + 4C^{opt} + 16 \cdot OPT = K(\hat{y}) + f^{opt} + 4C^{opt} + 16 \cdot OPT$. ◀

▶ **Corollary 6.5.** *There is a $(32 + \epsilon)$-approximation algorithm for the knapsack median problem.*

Acknowledgments I thank Deeparnab Chakrabarty for various stimulating discussions that eventually led to this work. I thank Chandra Chekuri for some useful discussions regarding the matroid-intersection median problem.

References

1 S. Ahmadian, Z. Friggstad, and C. Swamy. Local-search based approximation algorithms for mobile facility location problems. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1607–1621, 2013.

2 I. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. In *Proceedings of the 12th SODA*, pages 661–670, 2001.

3 I. Baev, R. Rajaraman, and C. Swamy. Approximation algorithms for data placement problems. *SIAM Journal on Computing*, 38(4):1411–1429, 2008.

4 D. Chakrabarty and C. Swamy. Facility location with client latencies: linear-programming based techniques for minimum latency problems. In *Proceedings of the 15th IPCO*, pages 92–103, 2011.

5 M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *JCSS*, 65(1):129–149, 2002.

6 M. Charikar and S. Li. A dependent LP-rounding approach for the k-median problem. In *Proceedings of the 39th ICALP*, pages 194–205, 2012.

7 F. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.

8 W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley and Sons, Inc., New York, 1998.

9 Z. Friggstad and M. Salavatipour. Minimizing movement in mobile facility location problems. *ACM Transactions on Algorithms*, 7(3), 2011.

10 I. Gørtz and V. Nagarajan. Locating depots for capacitated vehicle routing. In *Proceedings of the 14th APPROX*, pages 230–241, 2011.

11 R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha. The matroid median problem. In *Proceedings, 22nd SODA*, pages 1117–1130, 2011.

12 A. Kumar. Constant-factor approximation algorithm for the knapsack median problem. In *Proceedings of the 23rd SODA*, pages 824–832, 2012.

13 D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th STOC*, pages 265–274, 1997.

14 C. Swamy and D. B. Shmoys. Fault-tolerant facility location. In *Proceedings of the 14th SODA*, pages 735–736, 2003.