MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Facility Location with Matroid or Knapsack Constraints

Ravishankar Krishnaswamy
Computer Science Department, Princeton University

Amit Kumar
Department of Computer Science and Engg., IIT Delhi

Viswanath Nagarajan
Industrial and Operations Engineering Department, University of Michigan

Yogish Sabharwal
IBM India Research Laboratory

Barna Saha
AT&T Research

In the classical *k-median* problem, we are given a metric space and want to open k centers so as to minimize the sum (over all the vertices) of the distance of each vertex to its nearest open center. In this paper we present the first constant-factor approximation algorithms for two natural generalizations of this problem, that handle matroid or knapsack constraints.

In the *matroid median* problem, there is an underlying matroid on the vertices and the set of open centers is constrained to be independent in this matroid. When the matroid is uniform, we recover the k-median problem. Another previously studied special case is the *red-blue median* problem where we have a partition matroid with two parts. Our algorithm for matroid median is based on rounding a natural linear programming relaxation in two stages, and relies on a connection to matroid intersection.

In the *knapsack median* problem, centers have weights and the total weight of open centers is constrained to be at most a given capacity. When all weights are uniform, this reduces to the k-median problem. The algorithm for knapsack median is based on a novel LP relaxation that constrains the set of centers that each vertex can get connected to. The rounding procedure uses a two stage approach similar to that for matroid median.

Key words: Approximation Algorithms, Facility Location, Matroid, Knapsack
MSC2000 subject classification: 68W25, 90B80
OR/MS subject classification: Primary: analysis of algorithms; Secondary: network/graph algorithms

1. Introduction

The k-median problem is an extensively studied facility location problem. Given an n-vertex metric space (V,d) and a bound k, the goal is to locate/open k centers $C \subseteq V$ so as to minimize the sum of distances of each vertex to its nearest open center. The distance of a vertex to its closest open center is called its connection cost. Various constant-factor approximation algorithms are known for k-median [8, 16, 15, 7, 2], using different techniques such as LP rounding, primal-dual methods and local search. Currently, the best approximation ratio for k-median is $\sqrt{3}+1+\epsilon$ (for any fixed $\epsilon > 0$) due to Li and Svensson [19].

Variants and generalizations of the k-median problem occur in several applications, and hence it is of interest to obtain good algorithms for them. In this paper, we study two natural generalizations of k-median, namely under a *matroid* or *knapsack* constraint, and obtain the first constant-factor approximation algorithms for both.

We first study the *matroid median* problem, where the set of open centers has to form an independent set in a given matroid, with the objective of minimizing sum of connection costs. This formulation can capture several intricate constraints on the open centers, and contains as special cases: the classical k-median (uniform matroid of rank k), and an application in Content Distribution Networks described below.

Next, we consider the generalization of the k-median problem where each vertex has a weight associated with it and the total weight of open centers is required to be at most a bound B, i.e., the set of open centers should form a feasible packing in a knapsack of size B – this is the *knapsack median* problem. Clearly, the k-median problem is a special case where all weights are one and the knapsack capacity $B = k$.

As a consequence of our results, we also obtain the first constant-factor approximation algorithm for a problem arising in Content Distribution Networks (CDN), which was introduced by Hajiaghayi et al. [12]. Here, there are T different *types* of vertices and separate bounds $\{k_i\}_{i=1}^T$; the goal is to open at most k_i centers of each type-i so as to minimize the sum of connection costs. The vertex-types denote different types of servers in the Content Distribution Networks application. A constant-factor approximation algorithm for the $T = 2$ special case (called the *red-blue median* problem) was given in [12]. The algorithm for the case of general T follows immediately from our result on matroid median, by using a partition matroid with T parts.

1.1. Our Results and Techniques

Our first result is a 16-approximation algorithm for the matroid median problem. As noted above, this also implies the first constant factor approximation for the k-median problem with multiple (more than two) vertex-types [12]. Our algorithm for MatroidMedian is based on the natural LP-relaxation and is surprisingly simple. The main insight is in establishing a connection to matroid intersection. The algorithm computes an optimal LP solution and rounds it in two phases, the key points of which are described below.

- The first phase sparsifies the LP solution while increasing the objective value by a constant factor. This is somewhat similar to the LP-rounding algorithm for k-median in Charikar et al. [8]. However we cannot consolidate fractionally open centers as in [8]; this is because the open centers must additionally satisfy the matroid rank constraints. In spite of this, we show that the vertices and the centers can be clustered into disjoint 'star-like' structures.
- This structure ensured by the first phase of rounding allows us to define (in the second phase) another linear program, for which the sparsified LP solution is feasible, and has objective value at most $O(1)$ times the original LP optimum. Then we show that the second phase LP is in fact integral, via a relation to the matroid-intersection polytope. Finally, we re-solve the second phase LP to obtain an extreme point solution, which is guaranteed to be integral. This corresponds to a feasible solution to MatroidMedian of objective value $O(1)$ times the LP optimum.

Our second result concerns the *knapsack median* problem (also called weighted W-median in [12]), where vertices have weights and the open centers are required to satisfy a knapsack constraint. The objective is, as before, to minimize the total connection cost of all the vertices. We give a constant factor approximation algorithm for this problem as well. The main difficulty here lies in formulating a good LP relaxation since natural LPs for KnapsackMedian have unbounded integrality gaps.

Consider the LP relaxation (LP_1) in Section 3.1 (replacing the matroid rank constraint by a knapsack constraint) where $x(i,j)$ is 1 if vertex j is assigned to vertex i, and y_i is 1 if we locate a center at i. The following integrality gap example was given by Charikar and Guha [7] : there are two vertices of weight 1 and B respectively. The distance between them is D. Clearly, any integral solution can open only one center, and so costs D. Whereas the optimal fractional solution can open the weight 1 vertex integrally and the expensive vertex to an extent of $1 - \frac{1}{B}$, so the total cost will be $\frac{D}{B}$. In fact, we show that the integrality gap remains unbounded even if we strengthen the LP relaxation by adding "knapsack-cover" inequalities [6], that have been useful in reducing integrality gaps in problems involving knapsack constraints [4, 3].

One idea of getting around this problem would be to *augment* the LP relaxation with more information. Suppose we "guess" the maximum distance between a vertex and the center to which it gets assigned in an optimal solution – call this value L. We obtain a new LP relaxation $\mathsf{LP}_1(L)$ by setting $x(i,j)$ to 0 if $d(i,j) > L$. Then, $\min_{L\geq 0} \max\{\mathsf{LP}_1(L), L\}$ is a valid lower bound on the optimal KnapsackMedian value. This idea takes care of the above integrality gap example – if we set L to be a value less than D, $\mathsf{LP}_1(L)$ becomes infeasible (i.e. has optimal value ∞); and if $L > D$, we have D as a lower bound. But now, consider the same example as above where we have D vertices co-located at each of the two locations, and $D \gg B \gg 1$. Here, any integral solution will have cost at least D^2. However, $\min_{L\geq 0} \max\{\mathsf{LP}_1(L), L\} = \frac{D^2}{B}$ obtained by setting $L = D$. So this lower bound is also not adequate.

We use a different LP relaxation that relies on a lower bound which looks at *groups* of vertices rather than a single vertex. We show (in Section 4) that, based on a guess T of the optimal value, one can come up with a bound U_j for each vertex j, and write a new LP relaxation $\mathsf{LP}_4(T)$ that sets $x(i,j)$ to 0 if $d(i,j) > U_j$. These bounds U_js are non-uniform across vertices, and better than what can be obtained by just looking at a single vertex. Thus the lower bound for KnapsackMedian that we work with is $\min_{T\geq 0} \max\{\mathsf{LP}_4(T), T\}$. Our rounding algorithm, which goes along the same lines as that for the MatroidMedian problem, shows that the LP relaxation $\mathsf{LP}_4(T)$ has a constant integrality gap, except for one group of vertices. Our algorithm then assigns this residual group of vertices to a single open center and upper bounds the connection cost by $O(T)$. Altogether, the KnapsackMedian solution cost is bounded by a constant factor times the lower bound $\min_{T\geq 0} \max\{\mathsf{LP}_4(T), T\}$. We note that the actual constant in the approximation ratio turns out to be large, and we have not made an attempt to get the best value of this constant.

Remark: In a conference version of this paper [17], we also discussed the prize-collecting (or penalty) matroid median problem, and obtained a constant factor approximation algorithm for it by extending the algorithm for matroid median. However, we chose not to include that result here in order to focus only on our main techniques.

1.2. Related Work

The first approximation algorithm for the metric k-median problem was due to Lin and Vitter [20] who gave an algorithm that for any $\epsilon > 0$, produced a solution of objective at most $2(1+\frac{1}{\epsilon})$ while opening at most $(1+\epsilon)k$ centers; this was based on the filtering technique for rounding the natural LP relaxation. The first approximation algorithm that opened only k centers was due to Bartal [5], via randomized tree embedding and achieved an $O(\log n \cdot \log\log n)$-approximation ratio. Charikar et al. [8] obtained the first $O(1)$-approximation algorithm for k-median, by rounding the LP relaxation; they obtained an approximation ratio of $6\frac{2}{3}$. The approximation ratio was improved to 6 by Jain and Vazirani [16], using the primal dual technique. Charikar and Guha [7] further improved the primal-dual approach to obtain a 4-approximation algorithm. Later, Arya et al. [2] analyzed a natural local search algorithm that exchanges up to p centers in each local move, and proved a $3+\frac{2}{p}$ approximation ratio (for any constant $p \geq 1$). Recently, Gupta and Tangwongsan [11] gave a considerably simplified proof of the Arya et al. [2] result. It is known that the k-median problem on general metrics is hard to approximate to a factor better than $1+\frac{2}{e}$. On Euclidean metrics, the k-median problem has been shown to admit a PTAS by Arora et al. [1]. The most relevant paper to ours with regard to the rounding technique is Charikar et al. [8]: our algorithm builds on ideas used in their work.

Hajiaghayi et al. [12] introduced the red-blue median problem — where the vertices are divided into two categories and there are different bounds on the number of open centers of each type — and obtained a constant factor approximation algorithm. Their algorithm uses a local search using single-swaps for each vertex type. The motivation in [12] came from locating servers in Content Distribution Networks, where there are T server-types and strict bounds on the number of servers of each type. The red-blue median problem captured the case $T = 2$. It is unclear whether their approach can be extended to multiple server types, since the local search with single swap for each server-type has neighborhood size $n^{\Omega(T)}$. Furthermore, even a $(T-1)$-exchange local search has large locality-gap– see Appendix A. Hence it is not clear if local search can be applied to MatroidMedian, even in the case of a partition matroid. [12] also discussed the difficulty in applying the Lagrangian relaxation approach (see [16]) to the red-blue median problem; this is further compounded in the MatroidMedian problem since there are exponentially many constraints on the centers.

The KnapsackMedian problem admits a "bicriteria approximation" ratio via the filtering technique [20]. The currently best known tradeoff [7] implies for any $\epsilon > 0$, a $\left(1+\frac{2}{\epsilon}\right)$-approximation in the connection costs while violating the knapsack constraint by a *multiplicative* $(1+\epsilon)$ factor. Charikar and Guha [7] also showed that for each $\epsilon > 0$, it is not possible to obtain a trade-off better than $\left(1+\frac{1}{\epsilon},\, 1+\epsilon\right)$ relative to the natural LP relaxation. As mentioned in [12], an $O(\log n)$-approximation is achievable for $\mathsf{KnapsackMedian}$ (without violation of the knapsack constraint) via a reduction to tree-metrics, since the problem on trees admits a Polynomial Time Approximation Scheme.

Subsequent to the conference versions of this paper [17, 18], Charikar and Li [9] improved the constants in the approximation factor for both matroid and knapsack median (to 9 and 34 respectively), using an approach similar to ours, but with a more careful rounding. Very recently, Li and Svensson [19] obtained an improved $(1+\sqrt{3}+\epsilon)$-approximation algorithm for k-median (for any constant $\epsilon > 0$).

2. Preliminaries

The input to the $\mathsf{MatroidMedian}$ problem consists of a finite set of vertices V and a distance function $d: V \times V \to \mathbb{R}_{\geq 0}$ which is symmetric and satisfies the triangle inequality, i.e. $d(u,v)+d(v,w) \geq d(u,w)$ for all $u,v,w \in V$. Such a tuple (V,d) is called a finite metric space. We are also given a matroid $\mathcal{M}$, with ground set V and set of independent sets $\mathcal{I}(\mathcal{M}) \subseteq 2^V$. The goal is to open an independent set $S \in \mathcal{I}(\mathcal{M})$ of centers such that the sum $\sum_{u\in V} d(u,S)$ is minimized; here $d(u,S) = \min_{v\in S} d(u,v)$ is the connection cost of vertex u. We assume some familiarity with matroids, for more details see eg. [21].

The $\mathsf{KnapsackMedian}$ problem is similarly defined. We are given a finite metric space (V,d), non-negative weights $\{f_i\}_{i\in V}$ (representing facility costs) and a bound B. The goal is to open centers $S \subseteq V$ such that $\sum_{j\in S} f_j \leq B$ and the objective $\sum_{u\in V} d(u,S)$ is minimized.

3. Approximation Algorithm for Matroid Median

In this section, we describe a constant factor approximation algorithm for $\mathsf{MatroidMedian}$. We first give a natural LP relaxation for this problem, and then show how to round a fractional solution.

3.1. An LP Relaxation for $\mathsf{MatroidMedian}$

In the following linear program, y_v is the indicator variable for whether vertex $v \in V$ is opened as a center, and x_{uv} is the indicator variable for whether vertex u is served by center v. Then, the following LP is a valid relaxation for the $\mathsf{MatroidMedian}$ problem.

$$\text{minimize} \sum_{u\in V}\sum_{v\in V} d(u,v)x_{uv} \qquad (\mathsf{LP}_1)$$

$$\text{subject to} \sum_{v\in V} x_{uv} = 1 \qquad \forall\, u \in V \qquad (3.1)$$

$$x_{uv} \leq y_v \qquad \forall\, u \in V, v \in V \qquad (3.2)$$

$$\sum_{v\in S} y_v \leq r_{\mathcal{M}}(S) \qquad \forall\, S \subseteq V \qquad (3.3)$$

$$x_{uv}, y_v \geq 0 \qquad \forall\, u,v \in V \qquad (3.4)$$

If x_{uv} and y_v are restricted to only take values 0 or 1, then LP_1 can be seen to be an exact formulation for $\mathsf{MatroidMedian}$. The first constraint models the requirement that each vertex u must be connected to some center v, and the second one requires that it can do so only if the center v is opened, i.e. $x_{uv} = 1$ only if y_v is also set to 1. The constraints (3.3) are the matroid rank-constraint on the centers: they model the fact that the open centers form an independent set with respect to the matroid $\mathcal{M}$. Here $r_{\mathcal{M}} : 2^V \to \mathbb{Z}_{\geq 0}$ is the *rank-function* of the matroid, which is monotone and submodular. The objective value exactly measures the sum of the connection costs of each vertex. It is clear that given integrally open centers $y \in \{0,1\}^V$, each vertex $u \in V$ sets $x_{uv} = 1$ for its closest center v with $y_v = 1$. Let Opt denote an optimal solution of the given $\mathsf{MatroidMedian}$ instance, and let LP_1^* denote the LP optimum value. From the above discussion,

LEMMA 1. *The LP cost* LP_1^* *is at most the cost of an optimal solution* Opt.

3.2. Solving the LP: The Separation Oracle

Even though the LP relaxation has an exponential number of constraints, it can be solved in polynomial time (using the Ellipsoid method) assuming we can, in polynomial time, verify if a candidate solution (x,y)

satisfies all the constraints. Indeed, consider any fractional solution (x, y). Constraints (3.1), and (3.2) can easily be verified in $O(n^2)$ time, one by one.

Constraint (3.3) corresponds to checking if the fractional solution $\{y_v : v \in V\}$ lies in the matroid polytope for $\mathcal{M}$. Checking (3.3) is equivalent to seeing whether:

$$\min_{S \subseteq V} \left(r_{\mathcal{M}}(S) - \sum_{v \in S} y_v \right) \geq 0.$$

Since the rank-function $r_{\mathcal{M}}$ is submodular, so is the function $f(S) := r_{\mathcal{M}}(S) - \sum_{v \in S} y_v$. So the above condition (and hence (3.3)) can be checked using submodular function minimization, eg. [21, 13]. There are also more efficient methods for separating over the matroid polytope – refer to [21, 10] for more details on efficiently testing membership in matroid polyhedra. Thus we can obtain an optimal LP solution in polynomial time.

3.3. The Rounding Algorithm for $\mathsf{MatroidMedian}$

Let (x^*, y^*) denote the optimal LP solution. Our rounding algorithm consists of two stages. In the first stage, we only alter the x^*_{uv} variables such that the modified solution, while still being feasible to the LP, is also very sparse in its structure. In the second stage, we write another LP which exploits the sparse structure, for which the modified fractional solution is feasible with objective function at most a constant factor times LP^*_1. We then proceed to show that this new LP in fact corresponds to an integral polytope. Thus, by finding an optimal extreme point solution, we obtain an integral solution where the open centers form an independent set of $\mathcal{M}$, and the cost is $O(1)\mathsf{LP}^*_1$.

3.3.1. Stage I: Sparsifying the LP Solution

In the first stage, we follow the outline of the algorithm of Charikar et al. [8], but we cannot directly employ their procedure because in the matroid constrained setting we are not allowed to alter/consolidate the y^*_v variables in an arbitrary fashion. Specifically, step (i) below is identical to the first step (consolidating locations) in [8]. The subsequent steps in [8] do not apply since they consolidate centers; however using some ideas from [8] and with some additional work, we obtain the desired sparsification in steps (ii)-(iii) *without altering the y^*-variables*.

Step (i): Consolidating Clients. We begin with some notation, which will be useful throughout the paper. For each vertex u, let $\mathsf{LP}_u = \sum_{v \in V} d(u, v) x^*_{uv}$ denote the contribution of vertex u to the optimal LP value LP^*_1. Also, let $\mathcal{B}(u, r) = \{v \in V \mid d(u, v) \leq r\}$ denote the ball of radius r centered at vertex u. For any vertex u, we say that $\mathcal{B}(u, 2\mathsf{LP}_u)$ is the *local ball* centered at u.

Algorithm 1 Consolidating Clients for $\mathsf{MatroidMedian}$

1: initialize $w_u \leftarrow 1$ for all vertices $u \in V$, and $W \leftarrow \emptyset$.
2: order the vertices $u_1, u_2, \ldots, u_n$ according to non-decreasing LP_u values.
3: **for** $i \leftarrow 1, \cdots, n$ **do**
4: **if** there exists vertex $u_j \in W$ such that $d(u_i, u_j) \leq 4 \cdot \mathsf{LP}_{u_i}$ **then**
5: set $w_{u_j} \leftarrow w_{u_j} + 1$ and $w_{u_i} \leftarrow 0$.
6: **else**
7: set $W \leftarrow W \cup \{u_i\}$.
8: **end if**
9: **end for**
10: return clients $V' \leftarrow W$ and weights $\{w_u : u \in V'\}$.

In step 5, we can think of moving u_i to u_j for the rest of the algorithm (which is why we increase the weight of u_j and set the weight of u_i to be zero). Since $\mathsf{LP}_{u_i} \geq \mathsf{LP}_{u_j}$, this movement may only reduce the fractional value $\sum_u w_u \cdot \mathsf{LP}_u$. After the above process, V' denotes the set of vertices with positive weight, i.e. $V' = \{v \,|\, w_v > 0\}$. For the rest of the paper, we will refer to vertices in V' as *clients*. By the way we defined this set, it is clear that the following two observations hold.

CLAIM 1. *For $u, v \in V'$, we have $d(u,v) > 4 \cdot \max(\mathsf{LP}_u, \mathsf{LP}_v)$.*

This is true because, otherwise, if (without loss of generality) $\mathsf{LP}_v \geq \mathsf{LP}_u$ and $d(u,v) \leq 4\mathsf{LP}_v$, then we would have moved v to u when we were considering v. Claim 1 implies that the local balls $\mathcal{B}(u, 2\mathsf{LP}_u)$ centered at clients $u \in V'$ are disjoint.

CLAIM 2.

$$\sum_{u \in V'} w_u \sum_{v \in V} d(u,v) x^*_{uv} \quad \leq \quad \sum_{u \in V} \sum_{v \in V} d(u,v) x^*_{uv}$$

This is because, in the above reassignment process, the expression $\sum_{u \in V} w_u \sum_{v \in V} d(u,v) x^*_{uv}$ initially equals $\sum_{u \in V} \sum_{v \in V} d(u,v) x^*_{uv}$ and does not increase: when we move vertex u_i to u_j, a term corresponding to LP_{u_i} is replaced by an additional copy of LP_{u_j}, and $\mathsf{LP}_{u_i} \geq \mathsf{LP}_{u_j}$ (by the vertex ordering).

The following claim is a direct consequence of Markov's inequality.

CLAIM 3. *For any client $u \in V'$, $\sum_{v \in \mathcal{B}(u, 2\mathsf{LP}_u)} x_{uv} \geq 1/2$. In words, each client is fractionally connected to centers in its local ball to at least an extent of $1/2$.*

Finally, we observe that any solution to the new (weighted) instance is also a solution to the original instance, at a small increase in cost.

CLAIM 4. *For any $S \subseteq V$, $\sum_{v \in V} d(v,S) \leq \sum_{u \in V'} w_u \cdot d(u,S) + 4 \cdot \mathsf{LP}^*_1$.*

Proof. Define map $\tau : V \to V'$ as follows. If $v \in V'$ (i.e. step 7 occurs when $u_i = v$ in Algorithm 1) then $\tau(v) = v$. And if $v \in V \setminus V'$ (i.e. step 5 occurs when $u_i = v$ in Algorithm 1) then $\tau(v) = u_j$ is the vertex $u_j \in V'$ such that $d(v, u_j) \leq 4 \cdot \mathsf{LP}_v$ for which w_{u_j} is increased by one. By definition, we have $d(v, \tau(v)) \leq 4 \cdot \mathsf{LP}_v$. Notice also that $|\tau^{-1}(u)| = w_u$ for all $u \in V'$. Thus

$$\sum_{v \in V} d(v,S) \leq \sum_{v \in V} \left(d(v,\tau(v)) + d(\tau(v),S)\right) \leq 4 \cdot \sum_{v \in V} \mathsf{LP}_v + \sum_{u \in V'} |\tau^{-1}(u)| \cdot d(u,S) = 4 \cdot \mathsf{LP}^*_1 + \sum_{u \in V'} w_u \cdot d(u,S).$$

The first inequality is by triangle inequality, the second inequality is by $d(v,\tau(v)) \leq 4 \cdot \mathsf{LP}_v$ and the definition of map τ, and the last equality uses $|\tau^{-1}(u)| = w_u$. ■

We now assume that we have the weighted instance (with clients V'), and are interested in finding a set $S \subseteq V$ of centers to minimize $\sum_{u \in V'} w_u d(u,S)$. Note that centers may be chosen from the entire vertex-set V, and are not restricted to V'. Consider an LP-solution (x^A, y^*) to this weighted instance, where $x^A_{uv} = x^*_{uv}$ for all $u \in V'$, $v \in V$. Note that (x^A, y^*) satisfies constraints (3.1)-(3.2) with u ranging over V', and also constraint (3.3); so it is indeed a feasible fractional solution to the weighted instance. Also, by Claim 2, the objective value of (x^A, y^*) is $\sum_{u \in V'} w_u \sum_{v \in V} d(u,v) x^A_{uv} \leq \mathsf{LP}^*_1$, i.e. at most the original LP optimum.

After this step, even though we have made sure that the clients are well-separated, a client $u \in V'$ may be fractionally dependent on several partially open centers, as governed by the x_{uv} variables. More specifically, it may be served by the following types of centers: (i) contained in its local ball $\mathcal{B}(u, 2\mathsf{LP}_u)$, (ii) contained in another local ball $\mathcal{B}(u', 2\mathsf{LP}_{u'})$, and (iii) not contained in any of the local balls around clients. The subsequent steps further simplify the structure of these connections.

Remark: To illustrate the high-level idea behind our algorithm, suppose it is the case that for all $u \in V'$, client u is completely served by centers inside $\mathcal{B}(u, 2\mathsf{LP}_u)$. Then, we can infer that it is sufficient to open a center inside each of these balls, while respecting the matroid polytope constraints. Since we are guaranteed that for $u, v \in V'$, $\mathcal{B}(u, 2\mathsf{LP}_u) \cap \mathcal{B}(v, 2\mathsf{LP}_v) = \emptyset$ (from Claim 1), this problem reduces to that of finding an independent set in the intersection of *matroid* $\mathcal{M}$ and the *partition matroid* defined by the balls $\{\mathcal{B}(u, 2\mathsf{LP}_u) \,|\, u \in V'\}$! Furthermore, the fractional solution (x^*, y^*) is feasible for the natural LP-relaxation of the matroid intersection problem. Now, because the matroid intersection polytope is integral, we can obtain an integer solution of low cost (relative to LP^*_1).

However, the vertices may not in general be fully served by centers inside their corresponding local balls, as mentioned earlier. Nevertheless, we establish some additional structure (in the next three steps) which enables us to reduce to a problem (in Stage II) of intersecting matroid $\mathcal{M}$ with certain *laminar constraints* (instead of just partition constraints as in the above example).

Step (ii): Making the objective function uniform and centers private. We now simplify connections that any vertex participates outsides its local ball. We start with the LP-solution (x^A, y^*) and modify it to another solution (x^B, y^*). Initially set $x^B \leftarrow x^A$.

(A). For any client u that depends on a center v which is contained in another client u''s local ball, we change the coefficient of x_{uv} in the objective function from $d(u,v)$ to $d(u,u')$. Because the clients are well-separated, this changes the total cost only by a small factor. Formally,

$$\begin{aligned} d(u,v) &\geq d(u,u') - 2\mathsf{LP}_{u'} && \text{since } v \in \mathcal{B}(u', 2\mathsf{LP}_{u'}) \\ &\geq d(u,u') - d(u,u')/2 && \text{from Claim 1} \\ &\geq \frac{1}{2} \cdot d(u,u') && \end{aligned} \tag{3.5}$$

(B). We now simplify centers that are not contained in any local ball, and ensure that each such center has only one client dependent on it.

Consider any vertex $v \in V$ which does not lie in any local ball, and has at least two clients dependent on it. Let these clients be $u_0, u_1, \ldots, u_k$ ordered such that $d(u_0,v) \leq d(u_1,v) \leq \ldots \leq d(u_k,v)$. The following claim will be useful in this re-assignment.

CLAIM 5. *For all $i \in \{1, \ldots, k\}$, $d(u_i, u_0) \leq 2d(u_i, v)$.*

Proof. From the way we have ordered the clients, we know that $d(u_i,v) \geq d(u_0,v)$; so $d(u_i,u_0) \leq d(u_i,v) + d(u_0,v) \leq 2d(u_i,v)$ for all $i \in \{1, \cdots, k\}$. ■

Now, for each $1 \leq i \leq k$, we remove the connection (u_i, v) (ie. $x^B_{u_i v} \leftarrow 0$) and arbitrarily increase connections (for a total extent $x^A_{u_i v}$) to edges (u_i, v') for $v' \in \mathcal{B}(u_0, 2\mathsf{LP}_{u_0})$ while maintaining feasibility, i.e. $x^B_{u_i v'} \leq y^*_{v'}$). But we are ensured that a feasible re-assignment exists because for every client u_i, the extent to which it is connected outside its ball is at most $1/2$, and we are guaranteed that the total extent to which centers are opened in $\mathcal{B}(u_0, 2\mathsf{LP}_{u_0})$ is at least $1/2$ (Claim 3). Therefore, we can completely remove any connection u_i might have to v and re-assign it to centers in $\mathcal{B}(u_0, 2\mathsf{LP}_{u_0})$ and for each of these reassignments, we use $d(u_i, u_0)$ as the distance coefficient in the objective. Recall from Claim 5 that $d(u_i,u_0) \leq 2d(u_i,v)$.

After this step, only one client, u_0, depends on center v; all other clients $\{u_i\}_{i=1}^k$ which were originally connected to v, now have that connection reassigned among centers in $\mathcal{B}(u_0, 2\mathsf{LP}_{u_0})$. Now we say that v is a *private center* to client u_0.

We perform the above modification for all centers v that do not lie in any local ball. For any client u, let $\mathcal{P}(u)$ denote the set of all vertices that are either contained in $\mathcal{B}(u, 2\mathsf{LP}_u)$, or that are private to u. Notice that $\mathcal{P}(u) \cap \mathcal{P}(u') = \emptyset$ for any two clients $u, u' \in V'$. Also denote $\mathcal{P}^c(u) := V \setminus \mathcal{P}(u)$ for any $u \in V'$.

Notice that the edges modified in steps **(A)** and **(B)** above are disjoint. So the new objective function is:

$$\sum_{u \in V'} w_u \left[\sum_{v \in \mathcal{P}(u)} d(u,v) \cdot x^B_{uv} + \sum_{u' \in V' \setminus u} d(u,u') \sum_{v \in \mathcal{B}(u', 2\mathsf{LP}_{u'})} x^B_{uv} \right] \tag{3.6}$$

$$= \sum_{u \in V'} w_u \left[\sum_{v \in \mathcal{P}(u)} d(u,v) \cdot x^A_{uv} + \sum_{u' \in V' \setminus u} d(u,u') \sum_{v \in \mathcal{B}(u', 2\mathsf{LP}_{u'})} x^B_{uv} \right] \tag{3.7}$$

$$\leq \sum_{u \in V'} w_u \left[\sum_{v \in \mathcal{P}(u)} d(u,v) \cdot x^A_{uv} + \sum_{v \in V \setminus \mathcal{P}(u)} 2 \cdot d(u,v) \cdot x^A_{uv} \right] \quad \leq \quad 2 \cdot \mathsf{LP}^*_1. \tag{3.8}$$

Equality (3.7) is because we do not modify connections between any client u and centers in $\mathcal{P}(u)$. The first inequality in (3.8) is by (3.5) and Claim 5.

(C). We further change the LP-solution from (x^B, y^*) to (x^C, y^*) as follows. In x^C we ensure that any client which depends on centers in other clients' local balls, will in fact depend *only* on centers in the local ball of its nearest other client. For any client u, we reassign all connections (in x^B) to $\mathcal{P}^c(u)$ to centers of $\mathcal{B}(u', 2\mathsf{LP}_{u'})$ (in x^C) where u' is the closest other client to u. This is possible because the total reassignment for each client is at most half and every local-ball has at least half unit of centers (Claim 3). Clearly, the value of (x^C, y^*) under the new objective (3.6) is at most that of (x^B, y^*), since for any $u, u' \in V'$, all connections between client u and centers in $\mathcal{B}(u', 2\mathsf{LP}_{u'})$ have objective coefficient $d(u, u')$.

Now, for each $u \in V'$, if we let $\eta(u) \in V' \setminus \{u\}$ denote the closest other client to u, then u depends only on centers in $\mathcal{P}(u)$ and $\mathcal{B}\left(\eta(u), 2\mathsf{LP}_{\eta(u)}\right)$. Thus, the new objective value of (x^C, y^*) is exactly:

$$\sum_{u\in V'} w_u \left(\sum_{v\in\mathcal{P}(u)} d(u,v)x^C_{uv} + d(u,\eta(u))\cdot(1 - \sum_{v\in\mathcal{P}(u)} x^C_{uv}) \right) \quad \leq \quad 2\cdot\mathsf{LP}^*_1. \tag{3.9}$$

Observe that we retained for each $u \in V'$ only the x_{uv}-variables with $v \in \mathcal{P}(u)$; this suffices because all other x_{uw}-variables (with $w \in \mathcal{P}^c(u)$) have the same coefficient $d(u, \eta(u))$ in the objective, due to the changes made in steps **(A)**, **(B)** and **(C)**.

Furthermore, for any client u which depends on a private center $v \in \mathcal{P}(u) \setminus \mathcal{B}(u, 2\mathsf{LP}_u)$, it must be that $d(u,v) \leq d(u,\eta(u))$; otherwise, we can re-assign this (uv) connection to a center $v' \in \mathcal{B}(\eta(u), 2\mathsf{LP}_{\eta(u)})$ and improve in the (altered) objective function; again we use the fact that u might depend on $\mathcal{P}(u) \setminus \mathcal{B}(u, 2\mathsf{LP}_u)$ to total extent at most half and $\mathcal{B}(\eta(u), 2\mathsf{LP}_{\eta(u)})$ has at least half unit of open centers.

To summarize, the above modifications ensure that fractional solution (x^C, y^*) satisfies the following:

(i) For any two clients $u, u' \in V'$, we have $d(u,u') > 4\max(\mathsf{LP}_u, \mathsf{LP}_{u'})$. In words, this means that all clients are well-separated.

(ii) For each center v that does not belong to any ball $\{\mathcal{B}(u, 2\mathsf{LP}_u) : u \in V'\}$, only one client depends on it. That is, $\mathcal{P}(u) \cap \mathcal{P}(u') = \emptyset$ for all clients $u \neq u'$.

(iii) Each client u depends only on centers in its local ball, its private centers, and centers in the ball of its nearest other client. The extent to which it depends on centers of the latter two kinds is at most $1/2$.

(iv) If client u depends on a private center v, then $d(u,v) \leq d(u,u')$ for any other client $u' \in V'$.

(v) The total cost under the modified objective is at most $2 \cdot \mathsf{LP}^*_1$.

Step (iii): Building Small Stars. Let us modify the mapping η slightly:

• Index the clients in V' arbitrarily, and for each $u \in V'$ let $\eta(u) \in V' \setminus \{u\}$ denote the closest other client where ties are broken in favor of the client with smallest index.

• For each $u \in V'$, if it only depends (under LP solution x^C) on centers in $\mathcal{P}(u)$ (ie. centers in its local ball or its private centers) then reset $\eta(u) \leftarrow u$.

Consider a directed dependency graph with vertices V', having arc-set $\{(u, \eta(u)) | u \in V'\}$. It is easy to see (by the tie-breaking property above) that this directed graph only has cycles of length at most two (see Figure 3.1). We define all directed cycles in this dependency graph to be *pseudo-roots*. There are two types of pseudo-roots, consisting of (i) a single client u where $\eta(u) = u$, or (ii) a pair $\{u_1, u_2\}$ of clients where $\eta(u_1) = u_2$ and $\eta(u_2) = u_1$. Observe that the local balls of every pseudo-root contain at least a unit of open centers (according to fractional solution y^*), and each client has a directed path to some pseudo-root. So each component in the dependency graph is a directed tree along with a pseudo-root.

The procedure we describe next is similar to the reduction to "3-level trees" in [8]. We break up the directed trees into a collection of stars, by traversing the trees in a bottom-up fashion, going from the leaves to the root. This step of creating stars further simplifies the connections that each client u can have outside of $\mathcal{P}(u)$. For any arc (u, u'), we say that u' is the *parent* of u, and u is a *child* of u'. Any client $u \in V'$ with no in-arc is called a *leaf*.

Repeat the following as long as there is a non-leaf client which is not part of any pseudo-root. Let $u \in V'$ be a deepest such client[1] and u^{out} denote the parent of u.

1. Suppose there exists a child u_0 of u such that $d(u_0, u) \leq 2d(u, u^{\mathsf{out}}) = 2 \cdot d(u, \eta(u))$, then we make the following modification: let u_1 denote the child of u that is closest to u; we replace the directed arc (u, u^{out}) with (u, u_1), and declare the collection $\{u, u_1\}$ (which is now a 2-cycle), a pseudo-root. Observe that $d(u_0, u) \geq d(u, u^{\mathsf{out}})$ because u chose to direct its arc towards u^{out} instead of u_0.
2. If there is no such child u_0 of u, then for every child u^{in} of u, replace arc (u^{in}, u) with a new arc $(u^{\mathsf{in}}, u^{\mathsf{out}})$. In this process, u has its in-degree changed to zero, thereby becoming a leaf.

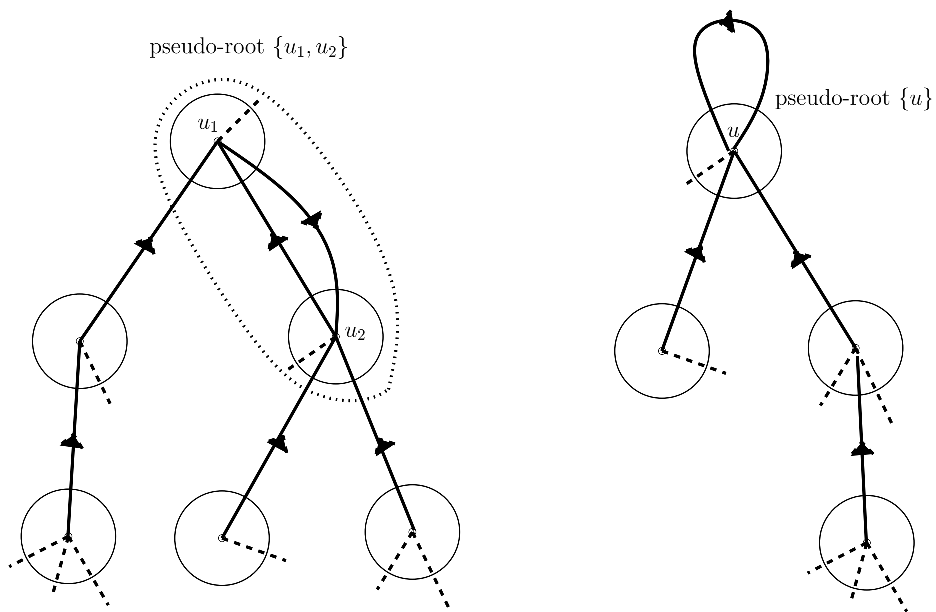

FIGURE 3.1. Solid arcs denote the dependency graph, circles are local balls around clients, dashed edges represent private centers.

Notice that we have maintained the invariant that there are no out-arcs from a pseudo-root to other clients, and every node has exactly one out-arc. Define mapping $\sigma : V' \to V'$ as follows: for each $u \in V'$, set $\sigma(u)$ to u's parent in the final dependency graph. Note that the final dependency graph is a collection of "stars" with centers as pseudo-roots, i.e. each client in V' has a path of length at most one to some pseudo-root.

CLAIM 6. *For each $w \in V'$, we have $d(w, \sigma(w)) \leq 2 \cdot d(w, \eta(w))$.*

Proof. The out-arc $(w, \eta(w))$ from w can change only in the following two cases.

(I) w is considered as client u in some iteration of the above procedure and step 1 applies. Then it follows that the out-arc of w is never changed after this step, and by definition of step 1, $d(w, \sigma(w)) \leq 2 \cdot d(w, \eta(w))$.

(II) $\eta(w)$ is considered as client u in some iteration of the above procedure and step 2 applies. In this step, the out-arc of w changes to $(w, \eta(\eta(w)))$, and it may change further if step 2 applies to $\eta(\eta(w))$ later. At the end of the above procedure, if $\sigma(w) = w'$ then we obtain that there is a directed path $\langle w = w_0, w_1, \cdots, w_t = w' \rangle$ in the initial dependency graph, i.e. $\eta(w_{i-1}) = w_i$ for all $i \in \{1, 2, \cdots, t\}$.

Let $d(w, \eta(w)) = d(w_0, w_1) = a$. We now show by induction on $i \in \{1, \cdots, t\}$ that $d(w_i, w_{i-1}) \leq a/2^{i-1}$. The base case of $i = 1$ is obvious. For any $i < t$, assuming $d(w_i, w_{i-1}) \leq a/2^{i-1}$, we will show that $d(w_{i+1}, w_i) \leq a/2^i$. Consider the point when w's out-arc is changed from (w, w_i) to (w, w_{i+1}); this must be so, since w's out-arc changes from (w, w_1) to (w, w_t) through the procedure. At this point, step 2 must have occurred at node w_i, and w_{i-1} must have been a child of w_i; hence $d(w_{i+1}, w_i) \leq \frac{1}{2} \cdot d(w_i, w_{i-1}) \leq a/2^i$.

[1] That is, u is not part of any pseudo-root and all children of u are leaves.

Thus we have $d(w,\sigma(w)) \le \sum_{i=1}^{t} d(w_i,w_{i-1}) \le a\sum_{i=1}^{t}\frac{1}{2^{i-1}} < 2a = 2\cdot d(w,\eta(w))$. ■

At this point, we have a fractional solution (x^C, y^*) that satisfies constraints (3.1)-(3.4) and:

$$\sum_{u\in V'} w_u \left[\sum_{v\in\mathcal{P}(u)} d(u,v)x^C_{uv} + d(u,\sigma(u))\cdot(1-\sum_{v\in\mathcal{P}(u)} x^C_{uv})\right] \quad\le\quad 4\cdot \mathsf{LP}_1^* \tag{3.10}$$

The inequality follows from (3.9) and Claim 6.

3.3.2. Stage II: Reformulating the LP

Based on the star-like structure derived in the previous subsection, we propose another linear program for which the fractional solution (x^C, y^*) is shown to be feasible with objective value as in (3.10). Crucially, we will show that this new LP is integral. Hence we can obtain an integral solution to it of cost at most $4\cdot\mathsf{LP}_1^*$. Finally, we show that any integral solution to our reformulated LP also corresponds to an integral solution to the original $\mathsf{MatroidMedian}$ instance, at the loss of another constant factor. Consider the LP:

$$\text{minimize} \sum_{u\in V'} w_u \quad \left[\sum_{v\in\mathcal{P}(u)} d(u,v)z_v \quad + \quad d(u,\sigma(u))\left(1-\sum_{v\in\mathcal{P}(u)} z_v\right)\right] \tag{LP_2}$$

$$\text{subject to} \sum_{v\in\mathcal{P}(u)} z_v \le 1 \qquad \forall\, u\in V' \tag{3.11}$$

$$\sum_{v\in\mathcal{P}(u_1)\cup\mathcal{P}(u_2)} z_v \ge 1 \qquad \forall\,\text{pseudo-roots } \{u_1,u_2\} \tag{3.12}$$

$$\sum_{v\in S} z_v \le r_{\mathcal{M}}(S) \qquad \forall\, S\subseteq V \tag{3.13}$$

$$z_v \ge 0 \qquad \forall\, v\in V \tag{3.14}$$

In constraint (3.12), we allow $u_1 = u_2$, which corresponds to a pseudo-root with a single client. The reason we have added the constraint (3.12) is the following. In the objective function, each client incurs only a cost of $d(u,\sigma(u))$ to the extent to which a private facility from $\mathcal{P}(u)$ is not assigned to it. This means that in our integral solution, we definitely want a facility to be chosen from the local balls of the pseudo-root to which u is connected if we do not open a private facility from $\mathcal{P}(u)$; this fact becomes clearer later. Also, this constraint does not increase the optimal value of the LP, as shown below.

CLAIM 7. *The linear program* LP_2 *has optimal value at most* $4\cdot\mathsf{LP}_1^*$.

Proof. Consider the solution z defined as: $z_v = \min\{y^*_v, x^C_{uv}\} = x^C_{uv}$ for all $v\in\mathcal{P}(u)$ and $u\in V'$; all other vertices have z-value zero. It is easy to see that constraints (3.11) and (3.13) are satisfied.

Constraint (3.12) is also trivially true for pseudo-roots $\{u\}$ consisting of only one client, since client u is fully served (in solution (x^C,y^*)) by centers in $\mathcal{P}(u)$. Now, let $\{u_1,u_2\}$ be any pseudo-root consisting of two clients. Recall that each $u\in\{u_1,u_2\}$ is connected to centers in ball $\mathcal{B}(u, 2\mathsf{LP}_u)\subseteq\mathcal{P}(u)$ to extent at least half; and as $\mathcal{P}(u_1)\cap\mathcal{P}(u_2)=\emptyset$ the total z-value inside $\mathcal{P}(u_1)\cup\mathcal{P}(u_2)$ is at least one. Thus z is feasible for LP_2, and by (3.10) its objective value is at most $4\cdot\mathsf{LP}_1^*$. ■

We show next that LP_2 is in fact, an integral polytope.

LEMMA 2. *Any basic feasible solution to* LP_2 *is integral.*

Proof. Consider any basic feasible solution z. Firstly, notice that the characteristic vectors defined by constraints (3.11) and (3.12) define a laminar family, since all the sets $\mathcal{P}(u)$ are disjoint.

Therefore, the subset of these constraints that are tightly satisfied by z define a laminar family. Also, by standard uncrossing arguments (see eg. [21]), we can choose the linearly-independent set of tight rank-constraints (3.13) to form a laminar family (in fact even a *chain*).

But then the vector z is defined by a constraint matrix which consists of *two laminar families* on the ground set of vertices. Such matrices are well-known to be *totally unimodular* [21]; eg. this fact is used in proving the integrality of the matroid-intersection polytope. This finishes the integrality proof. ■

It is clear that any integral solution feasible for LP_2 is also feasible for MatroidMedian, due to (3.13). We now relate the objective in LP_2 to the original MatroidMedian objective:

LEMMA 3. *For any integral solution $C \subseteq V$ to LP_2, the* MatroidMedian *objective value under solution C is at most 3 times its LP_2 objective value.*

Proof. We show that the connection cost of each client $u \in V'$ in MatroidMedian is at most 3 times that in LP_2. Suppose that $C \cap \mathcal{P}(u) \neq \emptyset$. Then u's connection cost is identical to its contribution to the LP_2 solution's objective. Therefore, we assume $C \cap \mathcal{P}(u) = \emptyset$.

Suppose that u is not part of a pseudo-root; let $\{u_1, u_2\}$ denote the pseudo-root that u is connected to. By constraint (3.12), there is some $v \in C \bigcap (\mathcal{P}(u_1) \cup \mathcal{P}(u_2))$. The contribution of u is $d(u, \sigma(u))$ in LP_2 and $d(u, v)$ in the actual objective function for MatroidMedian. We will now show that $d(u, v) \le 3 \cdot d(u, \sigma(u))$.

Without loss of generality let $\sigma(u) = u_1$ and suppose that $v \in \mathcal{P}(u_2)$; the other case of $v \in \mathcal{P}(u_1)$ is easier. From the property of private centers, we know $d(u_2, v) \le d(u_2, \eta(u_2)) \le d(u_2, u_1)$. Now if (u_1, u_2) is created as a new pseudo-root in step (iii).1, then we have the property that $d(u_1, u_2) \le d(u_1, u)$, since we choose the closest leaf to pair up with its parent to form the pseudo-root. Else (u_1, u_2) is an original pseudo-root, even before the modifications of step (iii). In that case, by definition $d(u_1, u_2) = d(u_1, \eta(u_1)) \le d(u_1, u)$. So, $d(u, v) \le d(u, u_1) + d(u_1, u_2) + d(u_2, v) \le d(u, u_1) + 2 \cdot d(u_1, u_2) \le 3 \cdot d(u, u_1) = 3 \cdot d(u, \sigma(u))$.

If u is itself a pseudo-root then it must be that $C \cap \mathcal{P}(u) \neq \emptyset$ by (3.12), contrary to the above assumption. If u is part of a pseudo-root $\{u, u'\}$. Then it must be that there is some $v \in C \cap \mathcal{P}(u')$, by (3.12). The contribution of u in LP_2 is $d(u, \sigma(u))$, and in MatroidMedian it is $d(u, v) \le d(u, u') + d(u', v) \le 2 \cdot d(u, u') = d(u, \sigma(u))$ (the second inequality uses the property of private centers). ■

To make this result algorithmic, we need to obtain in polynomial-time an extreme point solution to LP_2. Using the Ellipsoid method (as mentioned in Section 3.2) we can indeed obtain *some* fractional optimal solution to LP_2, which may not be an extreme point. However, such a solution can be converted to an extreme point of LP_2, using the method in Jain [14]. Due to the presence of both "$\le$" and "$\ge$" type constraints in (3.11)-(3.12) it is not clear whether LP_2 can be cast directly as an instance of matroid intersection.

Altogether, we obtain an integral solution to the weighted instance from **step (i)** of cost $\le 12 \cdot \mathsf{LP}_1^*$. Combined with Claim 4 from **step (i)**, we obtain:

THEOREM 1. *There is a 16-approximation algorithm for the matroid median problem.*

We have not tried to optimize the constant via this approach. However, getting the approximation ratio to match that for usual k-median would require additional ideas.

4. Approximation Algorithm for Knapsack Median

In this section we consider the KnapsackMedian problem. We are given a finite metric space (V, d), non-negative weights $\{f_i\}_{i \in V}$ and a bound B. The goal is to open centers $S \subseteq V$ such that $\sum_{j \in S} f_j \le B$ and the objective $\sum_{u \in V} d(u, S)$ is minimized.

The natural LP relaxation (LP_3) of this problem is similar to (LP_1) in Section 3.1, where we replace the constraint (3.3) with the knapsack constraint $\sum_{v \in V} f_v y_v \le B$. In addition, we can "guess" the maximum weight facility f_{max} used in an optimum solution, and if $f_v > f_{max}$ we set $y_v = 0$ (and hence $x_{uv} = 0$ as well, for all $u \in V$). This is clearly possible since there are only n different choices for f_{max}, and we can enumerate over them. Unfortunately, this LP_3 has an unbounded integrality gap. In Subsection 4.1, we show that a similar integrality gap persists even if we add *knapsack-cover (KC) inequalities* [6] to strengthen LP_3.

However, we show that there is a different LP relaxation which leads to a constant-factor approximation algorithm. This LP relaxation is based on a more careful pruning of which centers a vertex can get assigned to. Let $\mathtt{opt}$ denote the optimal value of the KnapsackMedian instance; we can enumerate over (polynomially many) values for T so that one of them satisfies $\mathtt{opt} \le T \le (1+\varepsilon)\mathtt{opt}$, for any fixed constant $\varepsilon > 0$. For each value of T, we will define an LP relaxation $\mathsf{LP}_4(T)$ and show that it can be rounded to produce a KnapsackMedian solution of cost at most $O(1) \cdot \mathsf{LP}_4^*(T) + O(1) \cdot T$, where $\mathsf{LP}_4^*(T)$ denotes the optimal value of $\mathsf{LP}_4(T)$. Moreover, for $T \approx \mathtt{opt}$ we show that $\mathsf{LP}_4^*(T) \le \mathtt{opt}$; so selecting the best KnapsackMedian solution over all choices of T would give us an $O(1)$-approximation algorithm.

The Linear Program $\mathsf{LP}_4(T)$**.** Given value T, this LP attempts to find a solution to $\mathsf{KnapsackMedian}$ of cost at most T. We may assume that in any $\mathsf{KnapsackMedian}$ solution, each vertex is assigned to its closest open center. So, if a vertex j is assigned to a center i, then any other vertex j' must be assigned to a center i' such that $d(i',j') \geq d(i,j) - d(j,j')$. Indeed, otherwise we might as well assign j to i' and reduce the total connection cost. So, the cost of the solution must be at least $\sum_{j' \in V} \max(0, d(i,j) - d(j,j'))$. If this value turns out to be greater than T, we know that j cannot be assigned to i in any $\mathsf{KnapsackMedian}$ solution of cost at most T. Thus, for each vertex $j \in V$, we define a bound U_j as the maximum value for which

$$\sum_{j' \in V} \max(0, U_j - d(j,j')) \quad \leq \quad T. \tag{4.15}$$

For a vertex j, we define the set of *admissible centers*, $\mathcal{A}(j)$, as the set of those vertices $i \in V$ such that $d(i,j) \leq U_j$. We now define the linear program:

$$\text{minimize} \sum_{u \in V} \sum_{v \in V} d(u,v) x_{uv} \tag{$\mathsf{LP}_4(T)$}$$

$$\text{subject to} \sum_{v \in V} x_{uv} = 1 \qquad \forall\, u \in V \tag{4.16}$$

$$x_{uv} \leq y_v \qquad \forall\, u \in V, v \in V \tag{4.17}$$

$$\sum_{v \in V} f_v \cdot y_v \leq B \tag{4.18}$$

$$x_{uv}, y_v \geq 0 \qquad \forall\, u, v \in V \tag{4.19}$$

$$x_{uv} = 0 \qquad \forall u \in V,\, v \notin \mathcal{A}(u) \tag{4.20}$$

From the above discussion, it follows that every $\mathsf{KnapsackMedian}$ solution of cost at most T remains feasible for $\mathsf{LP}_4(T)$. Hence for all $T \geq \mathtt{opt}$, we have $\mathsf{LP}_4^*(T) \leq \mathtt{opt}$.

We emphasize that even for $\mathsf{LP}_4(T)$, the integrality gap could be unbounded. However, the cost of our algorithm can be bounded by a constant factor times the optimal LP value $\mathsf{LP}_4^*(T)$, except for one group of vertices. For this group of vertices, we show that their connection cost is at most a constant times $\sum_{j' \in V} \max(0, U_j - d(j,j'))$ for some special vertex j, and hence, is at most $O(T)$.

The Rounding Algorithm for $\mathsf{KnapsackMedian}$**.** Let (x^*, y^*) be an optimal solution of $\mathsf{LP}_4(T)$. The rounding algorithm follows similar steps as in the $\mathsf{MatroidMedian}$ problem. The first stage is identical to Stage I of Section 3.3: modifying x_{uv} variables until we have a collection of disjoint stars centered at pseudo-roots. The total connection cost of the modified LP solution is $O(1) \cdot \mathsf{LP}_4^*(T)$. The sparsified solution satisfies the knapsack constraint since y_u variables are not modified in this stage. In the next stage II, we start with a new linear program $\mathsf{LP}_5(T)$ on variables $\{z_v : v \in V\}$ which is just LP_2 where constraint (3.13) is replaced by the knapsack constraint.

$$\text{minimize} \sum_{u \in V'} w_u \left[\sum_{v \in \mathcal{P}(u)} d(u,v) z_v + d(u, \sigma(u)) \left(1 - \sum_{v \in \mathcal{P}(u)} z_v \right) \right] \tag{$\mathsf{LP}_5(T)$}$$

$$\text{subject to} \sum_{v \in \mathcal{P}(u)} z_v \leq 1 \qquad \forall\, u \in V' \tag{4.21}$$

$$\sum_{v \in \mathcal{P}(u_1) \cup \mathcal{P}(u_2)} z_v \geq 1 \qquad \forall\, \text{pseudo-roots } \{u_1, u_2\} \tag{4.22}$$

$$\sum_{v \in V} f_v \cdot z_v \leq B \tag{4.23}$$

$$z_v \geq 0 \qquad \forall\, v \in V \tag{4.24}$$

However $\mathsf{LP}_5(T)$ is not integral as opposed to LP_2: it contains the knapsack problem as a special case. Instead, we give an *iterative relaxation* procedure (Algorithm 2) for rounding $\mathsf{LP}_5(T)$ to obtain the set of open centers $\mathcal{C} \subseteq V$.

Algorithm 2 Rounding Algorithm for KnapsackMedian $\mathsf{LP}_5(T)$

1: initialize $\mathcal{C} \leftarrow \emptyset$.
2: **while** $V \neq \emptyset$ **do**
3: find an extreme point optimum solution $\hat{z}$ to $\mathsf{LP}_5(T)$.
4: if there is a variable $\hat{z_v} = 0$, then remove variable $\hat{z_v}$, set $V \leftarrow V \setminus \{v\}$.
5: if there is a variable $\hat{z_v} = 1$, then $\mathcal{C} \leftarrow \mathcal{C} \cup \{v\}$, $V \leftarrow V \setminus \{v\}$ and $B \leftarrow B - f_v$.
6: **if** none of steps 4 and 5 holds, and $|V| \leq 2$ **then**
7: if $|V| = 1$, break.
8: if $|V| = 2$, open the center in V which has lesser weight; break.
9: **end if**
10: **end while**
11: return $\mathcal{C}$.

Before we argue about the correctness and approximation ratio of Algorithm 2, we note some useful properties of the solution produced by the Stage I rounding of Section 3.3. Recall that for a vertex u, $\mathcal{A}(u)$ denotes the set of admissible centers for u. Hence, if $x^*_{uv} > 0$ then $v \in \mathcal{A}(u)$. Recall that $\mathsf{LP}_u = \sum_v d(u,v) \cdot x^*_{uv}$; so $\mathsf{LP}^*_4(T) = \sum_{u \in V} \mathsf{LP}_u$. Also, V' is the set of vertices with positive weight after Step (i) of the Stage I rounding, and vertices in V' are referred to as clients.

LEMMA 4. *For any client $u \in V'$, $d(u,\sigma(u)) \leq 4 \cdot \max_{c \in \mathcal{A}(u)} d(u,c)$. Here $\sigma : V' \to V'$ is the map constructed in step (iii) of Stage I.*

Proof. The lemma is trivial if $\sigma(u) = u$. So, assume $\sigma(u) \neq u$. Consider step (ii) of the Stage I rounding – if we fractionally assign client u to centers in $\mathcal{P}(u'')$ for some $u'' \in V'$ (in the fractional solution x^B), then equation (3.5) and Claim 5 imply that $d(u,u'') \leq 2 \cdot d(u,c)$ for some $c \in \mathcal{A}(u)$. Hence, $d(u,\eta(u)) \leq 2d(u,c)$. Moreover, Claim 6 implies that $d(u,\sigma(u)) \leq 4 \cdot d(u,c)$. ■

For a client $u \in V'$, let $V_u \subseteq V$ be the set of vertices which got consolidated with u during Step (i) of the Stage I rounding; hence $|V_u| = w_u$. Note that for any vertex $s \in V_u$, $d(s,u) \leq 4 \cdot \mathsf{LP}_s$.

LEMMA 5. *For any client $u \in V'$:*

$$w_u \cdot d(u,\sigma(u)) \quad \leq \quad 64 \sum_{s \in V_u} \mathsf{LP}_s + 8 \cdot T.$$

Proof. Lemma 4 implies that there is an admissible center $c \in \mathcal{A}(u)$ such that $d(u,\sigma(u)) \leq 4 \cdot d(u,c)$. So it suffices to upper bound $w_u \cdot d(u,c) = \sum_{s \in V_u} d(u,c)$ by $16 \sum_{s \in V_u} \mathsf{LP}_s + 2 \cdot T$.

We divide the vertices in V_u into two sets: (i) V'_u containing those vertices s for which $d(s,c) \leq 3d(s,u)$, and (ii) V''_u containing those vertices s where $d(s,c) > 3d(s,u)$.

We deal with V'_u first. If $s \in V'_u$, then $d(u,c) \leq d(u,s) + d(s,c) \leq 4d(s,u) \leq 16 \cdot \mathsf{LP}_s$. Thus,

$$\sum_{s \in V'_u} d(u,c) \quad \leq \quad 16 \sum_{s \in V'_u} \mathsf{LP}_s.$$

Now, for any $s \in V''_u$, we have $d(u,c) \geq d(s,c) - d(s,u) \geq 2d(s,u)$. So $d(u,c) \leq 2(d(u,c) - d(s,u))$. Thus,

$$\sum_{s \in V''_u} d(u,c) \quad \leq \quad 2 \sum_{s \in V''_u} (d(u,c) - d(s,u)) \quad \leq \quad 2 \sum_{u' \in V} \max\{d(u,c) - d(u',u), 0\} \quad \leq \quad 2 \cdot T.$$

The last inequality uses the fact that $c \in \mathcal{A}(u)$ and (4.15). Thus $\sum_{s \in V_u} d(u,c) \leq 16 \sum_{s \in V_u} \mathsf{LP}_s + 2 \cdot T$. ■

The following lemma guarantees that the cost incurred by Algorithm 2 can be bounded in terms of the optimal values of the two LP relaxations $\mathsf{LP}_4(T)$ and $\mathsf{LP}_5(T)$.

LEMMA 6. *Algorithm 2 finds a feasible integral solution to* $\mathsf{KnapsackMedian}$ *having cost at most* $192 \cdot \mathsf{LP}_4^*(T) + 24 \cdot T + 3 \cdot \mathsf{LP}_5^*(T)$.

Proof. We first show that if the algorithm reaches step 11 then the solution $\mathcal{C}$ satisfies the knapsack constraint. In step 5, we always reduce the remaining budget by f_v whenever we add a center v to $\mathcal{C}$. The only other step that adds centers is step 8; we will show below that

$$\text{In step 8, } \hat{z}_{v_1} + \hat{z}_{v_2} = 1, \text{ where } V = \{v_1, v_2\}. \tag{4.25}$$

This would imply that we do not violate the knapsack constraint, as the smaller weight center is opened.

Now we show that the algorithm indeed reaches step 11. Steps 4 and 5 make progress in the sense that they reduce the number of variables: so there are at most $|V|$ iterations involving these two steps. Next, we show that for any extreme point solution $\hat{z}$ for which neither step 4 nor step 5 applies (i.e. there is no $z_v \in \{0,1\}$), step 6 applies and hence the algorithm reaches step 11. Let the linearly independent tight constraints defining $\hat{z}$ be indexed by $S \subseteq V'$ from (4.21), and R (pseudo-roots) from (4.22). From the laminar structure of the constraints and all right-hand-sides being 1, it follows that the sets corresponding to constraints in $S \bigcup R$ are all disjoint.[2] Furthermore, each set in $S \bigcup R$ contains at least two fractional variables. Hence, the number of positive variables in $\hat{z}$ is at least $2|S| + 2|R|$. Now, count the number of tight linearly independent constraints: there are at most $|S| + |R|$ tight constraints from (4.21)-(4.22), and one knapsack constraint (4.23). For an extreme point solution, the number of positive variables must equal the number of tight linearly independent constraints; so we obtain $|S| + |R| \le 1$ and that each set in $S \cup R$ contains exactly two vertices/variables. This is possible only when $|V| \le 2$; so step 6 applies. Thus the algorithm always reaches step 11, and it terminates.

Now we upper bound the connection cost of solution $\mathcal{C}$ and prove (4.25). There are three cases to consider:

(i) $S = R = \emptyset$ and $V = \{v\}$ with $v \in \mathcal{P}(u)$ for some $u \in V'$. In this case, step 8 does not occur. The $\mathsf{LP}_5(T)$ objective increases by at most $w_u \cdot d(u, \sigma(u)) \le 64 \cdot \mathsf{LP}_4^*(T) + 8 \cdot T$; the last inequality is by Lemma 5.

(ii) $S = \{u\}$, $R = \emptyset$ and $V = \{v_1, v_2\} \subseteq \mathcal{P}(u)$. Since the constraint corresponding to $u \in S$ is tight, we have $\hat{z}_{v_1} + \hat{z}_{v_2} = 1$. So (4.25) is satisfied. Say v_1 is the center opened in step 11. The increase in $\mathsf{LP}_5(T)$ objective is at most $w_u \cdot d(u, v_1) \le w_u \cdot d(u, \sigma(u)) \le 64 \cdot \mathsf{LP}_4^*(T) + 8 \cdot T$; the first inequality is because $v_1 \in \mathcal{P}(u)$ and the second is by Lemma 5.

(iii) $S = \emptyset$, $R = \{\langle u_1, u_2 \rangle\}$ and $V = \{v_1, v_2\} \subseteq \mathcal{P}(u_1) \cup \mathcal{P}(u_2)$. Again, by the tight constraint in R, we have $\hat{z}_{v_1} + \hat{z}_{v_2} = 1$, and (4.25) is satisfied. The increase in $\mathsf{LP}_5(T)$ objective (due to clients u_1 and u_2) is at most $\max\{w_{u_1} \cdot d(u_1, \sigma(u_1)),\, w_{u_2} \cdot d(u_2, \sigma(u_2))\} \le 64 \cdot \mathsf{LP}_4^*(T) + 8 \cdot T$ by Lemma 5.

The increase in the $\mathsf{LP}_5(T)$ objective due to all other iterations is zero since they do not change any variables. Thus the final objective value of integral solution $\mathcal{C}$ is at most $\mathsf{LP}_5^*(T) + 64 \cdot \mathsf{LP}_4^*(T) + 8 \cdot T$. Using Lemma 3 it follows that the connection cost in $\mathsf{KnapsackMedian}$ under this solution $\mathcal{C}$ is at most thrice its $\mathsf{LP}_5(T)$ objective. The lemma now follows. ■

Finally, by Claim 7, the Stage I rounding ensures that $\mathsf{LP}_5^*(T) \le 4 \cdot \mathsf{LP}_4^*(T)$. Combined with Lemma 6, Algorithm 2's solution to $\mathsf{KnapsackMedian}$ has cost at most $204 \cdot \mathsf{LP}_4^*(T) + 24 \cdot T$. Setting $T = \mathtt{opt}$, this implies a 228-approximation algorithm. Thus, we obtain:

THEOREM 2. *There is a constant factor approximation algorithm for the knapsack median problem.*

4.1. LP Integrality Gap for KnapsackMedian with Knapsack Cover Inequalities

In this section, we show that there is a large integrality gap for LP_3 even when strengthened with "Knapsack Cover" inequalities [6]. These inequalities have been useful in reducing integrality gaps in many settings involving knapsack constraints, eg. [4, 3].

First, we recall an integrality gap example for the basic LP_3.

[2] Recall that sets corresponding to constraints in S are of the form $\mathcal{P}(u)$ for some $u \in V'$, and sets of constraints in R are of the form $\mathcal{P}(u_1) \cup \mathcal{P}(u_2)$ for some pseudo-root $\{u_1, u_2\}$.

EXAMPLE 1 ([7]). Consider $|V| = 2$ with facility costs $f_1 = N$, $f_2 = 1$, distance $d(1,2) = D$ and bound $B = N$, for any large positive reals N and D. An integral solution that does not violate the knapsack constraint can open either center 1 or 2, but not both and hence has a connection cost of D. However, LP_3 can assign $y_1 = 1 - \frac{1}{N}$ and $y_2 = 1$ and thus has only D/N connection cost.

This bad example can be overcome by adding *knapsack covering (KC) inequalities* [6]. We now illustrate the use of KC inequalities in the KnapsackMedian problem. KC-inequalities are used for *covering* knapsack problems. Although KnapsackMedian has a packing constraint (at most B weight of open centers), it can be rephrased as a covering knapsack by requiring "at least $\sum_{v\in V} f_v - B$ weight of closed centers". Viewed this way, we can strengthen the basic LP_3 as follows.

Define for any subset of centers $S \subseteq V$, $f(S) := \sum_{v\in S} f(v)$. Then, to satisfy the knapsack constraint we need to close centers of weight at least $B' := f(V) - B$. For any subset $S \subseteq V$ of centers with $f(S) < B'$ we write the KC inequality:

$$\sum_{v\notin S} \min\{f(v), B' - f(S)\} \cdot (1 - y_v) \quad \geq \quad B' - f(S).$$

This inequality is valid for all integral solutions $\mathbf{y} \in \{0,1\}^V$ since, among centers in $V \setminus S$ at least $B' - f(S)$ weight must be closed. There are exponential number of such KC-inequalities; however using methods in [6] an FPTAS for the strengthened LP can be obtained. The addition of KC inequalities avoids the integrality gap in Example 1; there $B' = 1$ and setting $S = \emptyset$ yields:

$$\min\{1,1\} \cdot (1 - y_1) + \min\{1, N\} \cdot (1 - y_2) \geq 1,$$

ie. $y_1 + y_2 \leq 1$. Thus the LP optimum also has value D.

However the following example shows that the integrality gap remains high even with KC inequalities.

EXAMPLE 2. $V = \{a_i\}_{i=1}^R \bigcup \{b_i\}_{i=1}^R \bigcup \{p,q,u,v\}$ with metric distances d as follows: vertices $\{a_i\}_{i=1}^R$ (resp. $\{b_i\}_{i=1}^R$) are at zero distance from each other, $d(a_1,b_1) = d(p,q) = d(u,v) = D$ and $d(a_1,p) = d(p,u) = d(u,a_1) = \infty$. The facility costs are $f(a_i) = 1$ and $f(b_i) = N$ for all $i \in [R]$, and $f(p) = f(q) = f(u) = f(v) = N$. The knapsack bound is $B = 3N$. Moreover, $N > R \gg 1$.

Without loss of generality, an optimum integral solution opens exactly one center from each of $\{a_i\}_{i=1}^R \bigcup \{b_i\}_{i=1}^R$, $\{p,q\}$ and $\{u,v\}$ and hence has connection cost of $(R+2)D$.

On the other hand, we show that the KnapsackMedian LP_3 with KC inequalities has a feasible solution $\mathbf{z} \in [0,1]^V$ with much smaller cost. Define $z(a_i) = 1/R$ and $z(b_i) = \frac{N-1}{RN}$ for all $i \in [R]$, and $z(p) = z(q) = z(u) = z(v) = \frac{1}{2}$. Observe that the connection cost is $\left(\frac{R}{N} + 2\right) D < 3D$. Below we show that z is feasible; hence the integrality gap is $\Omega(R)$. The solution z clearly satisfies the constraint $\sum_{w\in V} f_w \cdot z_w \leq B$.

We now show that z satisfies all KC-inequalities. Note that $B' = f(V) - B = (R+1)N + R$ for this instance. Recall that KC-inequalities are written only for subsets S with $B' - f(S) > 0$. Also, KC-inequalities corresponding to subsets S with $B' - f(S) \geq N = \max_{w\in V} f_w$ reduce to $\sum_{w\notin S} f_w \cdot y_w \leq B$, which is clearly satisfied by z. Thus the only remaining KC-inequalities are from subsets S with $0 < B' - f(S) < N$, i.e. $f(S) \in [B' - N + 1, B' - 1] = [RN + R + 1, (R+1)N + R - 1]$. Since $R < N$, subset S must have at most $R+1$ cost N facilities. Thus there are at least three cost-N facilities H in $V \setminus S$. Since $z_w \leq \frac{1}{2}$ for all $w \in V$, we have $\sum_{w\in H}(1 - z_w) \geq \frac{3}{2}$. The KC-inequality from S is hence:

$$\begin{aligned}\sum_{w\notin S} \min\{f(w), B' - f(S)\}(1 - z_w) &\geq \sum_{w\in H} \min\{f(w), B' - f(S)\}(1 - z_w) \\ &= (B' - f(S)) \cdot \sum_{w\in H} (1 - z_w) \quad > \quad B' - f(S).\end{aligned}$$

The equality uses $B' - f(S) < N$ and that each facility-cost in H is N, and the last inequality is by $\sum_{w\in H}(1 - z_w) \geq \frac{3}{2}$ which was shown above.

5. Conclusion

In this paper, we studied two natural extensions of the k-median problem, when there is a matroid or knapsack constraint on the open centers. We obtained LP-based constant-factor approximation algorithms for both versions. It remains an interesting open question to obtain approximation ratios that match the best bound known for k-median.

References

[1] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for euclidean k-medians and related problems. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 106–113, 1998.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[3] N. Bansal, N. Buchbinder, and J. Naor. Randomized competitive algorithms for generalized caching. *SIAM J. Comput.*, 41(2):391–414, 2012.

[4] N. Bansal, A. Gupta, and R. Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1545, 2010.

[5] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[6] R. D. Carr, L. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 106–115, 2000.

[7] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4):803–824, 2005.

[8] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.

[9] M. Charikar and S. Li. A dependent LP-rounding approach for the k-median problem. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, pages 194–205, 2012.

[10] W. H. Cunningham. Testing membership in matroid polyhedra. *J. Comb. Theory, Ser. B*, 36(2):161–188, 1984.

[11] A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR*, abs/0809.2554, 2008.

[12] M. Hajiaghayi, R. Khandekar, and G. Kortsarz. Local search algorithms for the red-blue median problem. *Algorithmica*, 63(4):795–814, 2012.

[13] S. Iwata and J. B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 1230–1237, 2009.

[14] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[15] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 731–740, 2002.

[16] K. Jain and V. 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, 2001.

[17] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha. The matroid median problem. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 1117–1130, 2011.

[18] A. Kumar. Constant factor approximation algorithm for the knapsack median problem. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 824–832, 2012.

[19] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *CoRR*, abs/1211.0243, 2012.

[20] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inf. Process. Lett.*, 44(5):245–249, 1992.

[21] A. Schrijver. *Combinatorial optimization*. Springer, 2003.

Appendix A: Bad Example for Local Search with Multiple Swaps Here we give an example showing that any local search algorithm for MatroidMedian under a partition matroid of T parts that uses at most $T-1$ swaps cannot give an approximation factor better than $\Omega(\frac{n}{T})$; here n is the number of vertices.

The metric is uniform on $T+1$ locations. There are two servers of each type: each location $\{2,3,\ldots,T\}$ contains two servers; locations 1 and $T+1$ contain a single server each. For each $i \in \{1,\cdots,T\}$, the two copies of server i are located at locations i (first copy) and $i+1$ (second copy). There are $m \gg 1$ clients at each location $i \in \{1,\cdots,T\}$ and just one client at location $T+1$; hence $n = 2T + mT + 1$. The bounds on server-types are $k_i = 1$ for all i. The optimum solution is to pick the *first copy* of each server type and thus pay a connection cost of 1 (the client at location $T+1$). However, it can be seen that the solution consisting of the *second copy* of each server type is locally optimal, and its connection cost is m (clients at location 1). Thus the locality gap is $m = \Omega(n/T)$.