Constant Factor Approximation Algorithm for the Knapsack Median Problem

Amit Kumar *

Abstract

We give a constant factor approximation algorithm for the following generalization of the k-median problem. We are given a set of clients and facilities in a metric space. Each facility has a facility opening cost, and we are also given a budget B. The objective is to open a subset of facilities of total cost at most B, and minimize the total connection cost of the clients. This settles an open problem of Krishnaswamy-Kumar-Nagarajan-Sabharwal-Saha. The natural linear programming relaxation for this problem has unbounded integrality gap. Our algorithm strengthens this relaxation by adding constraints which stipulate which facilities a client can get assigned to. We show that after suitably modifying a fractional solution, one can get rich structural properties which allow us to get the desired approximation ratio.

1 Problem Definition

The problem of locating facilities to service a set of demands has been widely studied in computer science and operations research communities [LMW98, MF90]. The trade-off involved in such problems is the following – we would like to open as few facilities as possible, but the clients should not be located too far from the nearest facility. The k-median problem balances the two costs as follows : we are given a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of potential facilities lying in a metric space. The goal is to open at most k facilities in $\mathcal{F}$ so that the average distance traveled by a client in $\mathcal{D}$ to the nearest open facility is minimized.

The k-median problem is one of the most well-studied facility location problems with several constant factor approximation algorithms [AGK$^+$01, CG99, CGTS02, JV01]. Motivated by applications in content distribution networks, Hajiaghayi et al. [HKK10] considered the following generalization of the k-median problem, which they called the *Red-Blue Median Problem* – the set of facilities are partitioned into two sets – $\mathcal{F}_1$ and $\mathcal{F}_2$, and we are given two parameters k_1 and k_2. The goal is to open at most k_1 facilities of $\mathcal{F}_1$ and k_2 facilities of $\mathcal{F}_2$ such that the total connection cost of the clients is minimized. They gave a constant factor approximation algorithm for this problem. Krishnaswamy et al. [KKN$^+$11] generalized this result to the case of arbitrary number of partitions of $\mathcal{F}$. In fact, their result holds even when the set of open facilities is required to be an independent set in a matroid (the *matroid median problem*). They show that the natural linear programming relaxation for this problem has constant integrality gap.

In this paper, we consider the following problem. As in the k-median problem, we are given a set of clients $\mathcal{D}$ and facilities $\mathcal{F}$ in a metric space. Each client j has an associated demand d_j, each facility i has a facility opening cost f_i and we are given a budget B. The goal is to open a set of facilities such that their total opening cost is at most B, and minimize the total connection cost of the clients, i.e., $\sum_{j \in \mathcal{D}} d_j c(i(j), j)$, where $i(j)$ is the facility to which j gets assigned, and c denotes the distance in the underlying metric space. We call this the `Knapsack Median Problem`. Clearly, the k-median problem is a special case of this problem where all facilities costs are one, and $B = k$. In this paper, we give a constant factor approximation algorithm for the `Knapsack Median Problem`. This answers an open question posed by [KKN$^+$11].

The main difficulty here lies in the fact the natural LP relaxation has unbounded integrality gap. This happens even when all facility costs are at most B (the natural LP relaxation for the knapsack problem also has unbounded integrality gap, but it becomes a constant if we remove all items of size more than the knapsack capacity). Consider the LP relaxation given in Section 3 where $x(i, j)$ is 1 if client j is assigned to facility i, and y_i is 1 if facility i gets opened. The following integrality gap example was given by Charikar and Guha [CG05] : there are two facilities of cost 1 and B respectively, and two clients (with unit demand) co-located with the two facilities respectively. The distance between the two facilities is a large number D. Clearly, any integral solution can open only one facility, and so must pay D, whereas the optimal fractional solution can open the expensive facility to an extent of $1 - \frac{1}{B}$, and so the total

*Dept. of Computer Science and Engg., IIT Delhi, India-110016, email : `amitk@cse.iitd.ac.in`

cost will be $\frac{D}{B}$. Krishnaswamy et al. [KKN+11] showed that the integrality gap remains unbounded even if we strengthen the LP relaxation by adding knapsack-cover inequalities.

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 client and the facility to which it gets assigned in an optimal solution – call this value L. In the LP relaxation, we can set $x(i, j)$ to 0 if $c(i, j) > L$. This would take care of the above integrality gap example – if we set L to be a value less than D, the LP becomes infeasible, and if $L > D$, we already have D as a lower bound because we have guessed that at least one demand has connection cost at least D in the optimal solution. But now, consider the same example as above where we have D clients located at each of the two facilities respectively. Now, any integral solution will have cost at least D^2, and even if we plug in $L > D$, the LP can get away with value $\frac{D^2}{B}$ only. The lower bound of D is also not enough. Therefore, we need a more subtle way of coming up with a lower bound which looks at *groups* of clients rather than a single client. We show that, based on a guess of the value of the optimal solution, one can come up with lower bounds U_j for each client j, and set $x(i, j)$ to 0 in the LP relaxation if $d(i, j) > U_j$. Further these lower bounds are better than what one can obtain by just looking at client j alone. Our rounding algorithm, which closely follows that of Krishnaswamy et al. [KKN+11], shows that the natural LP relaxation (where we use the bounds U_j as mentioned) has constant integrality gap except for one group of demands. Our algorithm assigns this group of demands to a single open facility and the connection cost can be bounded by the value of the optimal solution (if our guess for this value is correct). Note that the actual constant in the approximation ratio turns out to be large, and we have not made an attempt to get the optimal value of this constant by balancing various parameters.

1.1 Related Work

The k-median problem has been extensively studied in the past and several constant factor approximation algorithms are known for this problem. Lin and Vitter [LV92] gave a constant factor approximation algorithm for this problem while opening at most $k(1 + \varepsilon)$ facilities for an arbitrarily small positive constant ε, even when distances do not obey triangle inequality. Assuming that distances obey triangle inequality, the first constant factor approximation algorithm was given by Charikar et al. [CGTS02]. Jain and Vazirani [JV01] gave a primal-dual constant factor approximation algorithm for this problem. Their algorithm first gives a primal-dual algorithm for the facility location problem which has the Lagrange multiplier preserving property (see e.g. [Mes07]). However, their algorithm does not extend to our problem. Indeed, if we use their approach, then we would get two solutions – one of these would open facilities which cost less than the budget B and the other one would spend more than B. Since facilities have non-uniform costs, the idea of combining these two solutions using a randomized algorithm does not seem to work here.

There are several approximation algorithms based on local search techniques as well [KPR98, AGK+01]. Hajiaghayi et al. [HKK10] used this approach to get a constant factor approximation algorithm for the case of red-blue median problem – recall that here there are two kinds of facilities (red and blue), and for each kind, we have a bound on the number of facilities that can be opened. Each operation in these local search algorithms swaps only one facility at a time. Since facilities have costs, we may need to open and close multiple facilities in each operation. It remains a challenge to analyze such a local search algorithm.

Krishnaswamy et al. [KKN+11] gave a constant factor approximation algorithm for the matroid median problem. Here, the set of open facilities should form an independent set in a given matroid. A natural special case (and in fact, this captures many of the ideas in the algorithm) is when the set of facilities is partitioned into K groups, and we are given an upper bound on the number of open facilities of each group. They show that the natural LP relaxation has constant integrality gap. Their algorithm begins by using ideas inherent in the algorithm of Charikar et al. [CGTS02], but has more subtle details. In fact, they also give a constant factor approximation for the `Knapsack Median Problem`, but exceed the budget B by the maximum cost of any facility. Our rounding algorithm also proceeds along the same lines as the latter algorithm, but the presence of the non-uniform bounds U_j allow us to avoid exceeding the budget B.

There are several bi-criteria approximation algorithms for the `Knapsack Median Problem` which violate the budget by $(1 + \varepsilon)$-factor for any $\varepsilon > 0$, and come within a constant of the total connection cost [LV92, CG05]. As mentioned above, Krishnaswamy et al. [KKN+11] also gave a constant factor approximation algorithm for this problem while violating the budget by at most the maximum cost of a facility.

1.2 Our Techniques

Consider the natural linear programming relaxation given in Section 3. As explained in the previous section, the integrality gap of this relaxation is unbounded. Now, suppose we know (up to a constant factor) the value of the optimal solution – call this OPT (we can do this by binary search). Based on this guess, we can come up with a bound U_j for each client j as follows. Suppose j is a assigned to a facility i where $c(i, j)$ is at least a parameter U_j. Then any other client j' must be assigned to a facility i' satisfying $c(i', j') \geq U_j - c(j, j')$ distance away from it (otherwise we can improve the connection cost of j). Hence, we can deduce that $\sum_{j' \in \mathcal{D}} d_{j'} \max(0, U_j - c(j, j')) \leq \texttt{OPT}$. We set U_j to be the highest value which satisfies this condition. In the LP relaxation, we set $x(i, j)$ to 0 if $c(i, j) > U_j$.

We now briefly describe the rounding algorithm. It proceeds along the same lines as that of Krishnaswamy et al. [KKN+11], but we give the details for the sake of completeness. Consider a fractional solution (x, y) to the LP relaxation. For a client j, the fractional solution assigns it fractionally to several facilities. Let $\Delta(j) = \sum_{i \in \mathcal{F}} c(i, j)x(i, j)$ denote the average distance to the facilities to which j gets fractionally assigned. Using the ideas in [CGTS02], we can assume (with constant loss in approximation ratio) that the distance between two different clients j and j' is at least a large constant times $\max(\Delta(j), \Delta(j'))$. This is achieved by merging several clients into a new client whose demand is the sum of demands of these clients. Let $B(j)$ denote the ball of radius $2\Delta(j)$ around client j. It is easy to check by a simple averaging argument that at least half of demand of j is assigned to facilities in $B(j)$. Also, these balls are disjoint (in fact, far from each other).

For each client j, we define another ball $E(j)$ – it is the set of facilities which are closer to j than to other clients. Clearly, the balls $E(j)$ are disjoint, and it is easy to check that $B(j)$ is a subset of $E(j)$ (see Figure 1). Krishnaswamy et al. [KKN+11] showed that, up to a constant loss in approximation ratio, a fractional solution can be massaged to have the following structure :

- A client j is fractionally assigned to facilities in $E(j)$ and the facilities in at most one of the balls $B(j')$, for some client j'.
- We can construct a directed graph G as follows – the set of vertices is same as the set of clients, and we have an arc (j, j') if j' is fractionally assigned to a facility in $B(j')$. This graph has the nice property that each of its components is a directed star where the root is either a single vertex or a 2-cycle.

Once we have established these properties, we write a new linear programming relaxation which enforces such properties. This is similar to the new linear programming relaxation in [KKN+11]. For the case of the matroid median problem, they showed that this new relaxation is totally unimodular and so they could directly recover a good integral solution from such a relaxation. For the case of `Knapsack Median Problem`, they used the iterative rounding method. They showed that as long as there are more than two facilities, one can always find a facility which is integrally open or closed in a vertex solution. When there are only two facilities, they just open both the facilities thus violating the budget constraint. For us, this is precisely where the presence of the bounds U_j helps. However, instead of giving an algorithm based on iterative rounding (which would require finding several vertex solutions), we show that one can directly argue about a vertex solution to this new LP relaxation. We show that in a vertex solution, all facilities are integrally open except for the facilities in the set $E(j)$ for at most *two* special clients j. One can easily account for the connection cost of all the clients except for these special clients. We connect these special clients to any open facility to which they were fractionally assigned in the LP solution. Now, note that such a client was perhaps obtained by merging several clients of $\mathcal{D}$ in the beginning of the rounding procedure, and so the connection cost of j really corresponds to the connection cost of several clients. We would like to bound the connection cost of all of these clients by the optimal value. This is where we use the property of the upper bounds $U_{j'}$ for clients $j' \in \mathcal{D}$.

In Section 2, we formally define the problem. In Section 3, we give the LP relaxation and define the bounds U_j for each client j. We first describe how to massage a fractional solution to make it well-structured in Section 4.1 – as mentioned earlier, this is same as the algorithm of Krishnaswamy et al. [KKN+11], though the notation is slightly different. In Section 4.2, we write a new LP relaxation, which is again based on [KKN+11], which actually enforces such a structure. Then we show that any vertex solution to this relaxation is almost integral, and so one can get a constant factor approximation algorithm.

2 Preliminaries

We are given a set of clients $\mathcal{D}$ and a set of potential facilities $\mathcal{F}$. Further, these points lie in a metric space, where we denote the distance between two points i and j by $c(i, j)$. Each facility i has a facility opening cost f_i, and we are given a budget B. Each client $j \in D$ has an associated value d_j, which denotes the amount of demand it has. Therefore, an instance $\mathcal{I}$ can be described by a tuple $(\mathcal{D}, \mathcal{F}, c, f, d)$. A solution opens a subset of facilities $\mathcal{F}'$ of total cost at most the budget B, and assigns each client in $\mathcal{D}$ to the closest open facility (in the set $\mathcal{F}'$). The objective is to minimize the total connection cost, i.e., $\min \sum_{j \in \mathcal{D}} d_j \cdot c(i(j), j)$, where $i(j)$ is the facility to which j gets assigned. In rest of the paper, we shall use i to refer to a facility and j to refer to a client.

3 LP Relaxation

Fix an instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, c, f, d)$. We have the following LP relaxation, which we call $\mathtt{LP}(\mathcal{I})$. Here, $x(i,j)$ is 1 iff client j is assigned to facility i, and y_i is 1 iff facility i is opened. The first constraint states that a client must be assigned to a facility, and the second constraint states that if a client is assigned to a facility, then the facility must be open. The third constraint requires that we do not exceed the budget B.

$$
\begin{aligned}
\min \sum_{j \in \mathcal{D}} d_j \cdot & \sum_{i \in \mathcal{F}} c(i,j) x(i,j) \\
\sum_{i \in \mathcal{F}} x(i,j) &= 1 \qquad \text{for all } j \in \mathcal{D} \\
x(i,j) &\le y_i \qquad \text{for all } i \in \mathcal{F}, j \in \mathcal{D} \\
\sum_{i \in \mathcal{F}} f_i y_i &\le B \\
x(i,j), y_i &\ge 0
\end{aligned}
$$

As argued above, this relaxation may have unbounded integrality gap. So we need to strengthen the relaxation. Suppose we know the value of the optimal solution – call it $\mathtt{OPT}(\mathcal{I})$ (we can guess this up to a factor $(1+\varepsilon)$ for an arbitrarily small constant ε by binary search). Among all the open facilities in the optimal solution, a client should be assigned to the closest one. So, if a client j is assigned to a facility i, then any other client j' must be assigned to a facility i' such that $c(i',j') \ge c(i,j) - c(j,j')$ away from 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 $\sum_{j' \in \mathcal{D}} d_{j'} \max(0, c(i,j) - c(j,j'))$. If this value turns out to be greater than $\mathtt{OPT}(\mathcal{I})$, we know that the optimal solution cannot assign j to i. Thus, for each client j, we define a bound U_j as the maximum value for which

$$\sum_{j' \in \mathcal{D}} d_{j'} \max(0, U_j - c(j,j')) \le \mathtt{OPT}(\mathcal{I}).$$

For a client j, we define the set of *allowable facilities*, $\mathcal{A}(j)$, as the set of those facilities i such that $c(i,j) \le U_j$. Hence, we can make the following modification to $\mathtt{LP}(\mathcal{I})$: we set $x(i,j)$ to 0 if $i \notin \mathcal{A}(j)$. We emphasize that even with this modification, the integrality gap of the LP could be unbounded. However, if our guess of $\mathtt{OPT}(\mathcal{I})$ is correct, then the cost of the solution produced by our algorithm can be bounded by the LP value except for one group of demands. For these group of clients, we show that their connection cost is at most a constant times $\sum_{j' \in \mathcal{D}} d_{j'} \max(0, U_j - c(j,j'))$ for some client j, and hence, is at most $O(\mathtt{OPT}(\mathcal{I}))$.

4 Rounding a fractional solution

We start with a fractional solution (x,y) to $\mathtt{LP}(\mathcal{I})$. There are several conceptual steps in the rounding algorithm. First, we modify the fractional solution through a sequence of steps to a new fractional solution which has a much cleaner structure – as mentioned earlier, this is same as that of [KKN+11]. Using this structure, we write a new LP relaxation for the problem. Finally, we show that a vertex solution of this new LP relaxation turns out to have some nice properties which can be exploited for rounding it to an integral solution.

4.1 Modifying the LP solution

(i) Consolidating clients : In this step (which is also used by Charikar et al. [CGTS02]), we shall merge some of the clients into a single client. We shall do this formally by updating the demands d_j of the clients. The new demand of a client j will be denoted by $d_j^{(1)}$. Initially, $d_j^{(1)} = d_j$ for all j.

For a client j, let $\Delta(j)$ denote $\sum_i c(i,j)x(i,j)$, the fractional distance traveled by j. Consider the clients in increasing order of the $\Delta(j)$ values – let this ordering be $j_1, \ldots, j_n$. When we consider j_l, if there exists a client j_u, $u < l$, such that $d_{j_u}^{(1)} > 0$ and $c(j_l, j_u) \le 4\Delta(j_l)$, then we increase $d_{j_u}^{(1)}$ by d_{j_l}, and set $d_{j_l}^{(1)}$ to 0. In other words, we move j_l to j_u and merge the two demands. Let $\mathcal{D}^{(1)}$ denote the new set of clients with non-zero demands. The following observations are easy to check.

Fact 4.1. *[CGTS02] If $j_1, j_2 \in \mathcal{D}^{(1)}$, then $c(j_1, j_2) \ge 4\max(\Delta(j_1), \Delta(j_2))$. Further,*

$$\sum_{j \in \mathcal{D}^{(1)}} d_j^{(1)} \Delta(j) \le \sum_{j \in \mathcal{D}} d_j \Delta(j).$$

(ii) Consolidating facilities : For a client $j \in \mathcal{D}^{(1)}$, define $B(j)$ as the ball of radius $2\Delta(j)$ around j. Fact 4.1 shows that these balls are disjoint. It is also easy to see, using a simple averaging argument, that

$$\sum_{i \in B(j)} y_i \ge 1/2.$$

We define a new solution $(x^{(1)}, y^{(1)})$ as follows. For each client j, we perform the following steps : let $i^\star$ be the cheapest facility in $B(j)$. We open a new facility $\mathtt{cen}(j)$

at the same location as j. The facility opening cost of $\texttt{cen}(j)$ is same as that of $i^\star$. We close all the facilities in $B(j)$, and set $y^{(1)}_{\texttt{cen}(j)}$ to be equal to $\min(1, \sum_{i \in B(j)} y_i)$. In other words, we are moving all the facilities in $B(j)$ to $\texttt{cen}(j)$. It is easy to check that the total facility opening cost does not increase. We need to change the variables $x(i', j')$ as well. We set $x^{(1)}(i', j')$ to 0 if $i' \in B(j) \setminus \{\texttt{cen}(j)\}$ for some j, and correspondingly increase $x^{(1)}(\texttt{cen}(j), j')$. In other words, for every pair of clients j and j', we define

$$x^{(1)}(\texttt{cen}(j), j') = \sum_{i \in B(j)} x(i, j).$$

Thus, the only open facility inside a ball $B(j)$ is the facility $\texttt{cen}(j)$. Let $\mathcal{F}^{(1)}$ be the set of facilities with non-zero $y^{(1)}_i$ values – these will either be $\texttt{cen}(j)$ for some client j, or the facilities in $\mathcal{F}$ which did not lie in any of the balls $B(j)$. Again, it is easy to check the following fact.

Fact 4.2. *For any $j \in \mathcal{D}^{(1)}$, $y^{(1)}_{\texttt{cen}(j)} \geq 1/2$. Further, if $x(i, j) > 0$ for some $i \in B(j'), j' \in \mathcal{D}^{(1)}$, then $c(\texttt{cen}(j'), j) \leq 2c(i, j)$. Hence, the cost of the solution $(x^{(1)}, y^{(1)})$ is at most twice that of (x, y).*

Proof: Fix a client j. The connection cost of j to facilities in $B(j)$ only decreases because $c(j, \texttt{cen}(j)) = 0$. Now, suppose j was assigned to a facility $i \in B(j')$ in the solution (x, y). In the new solution, it will get assigned by the same extent to $\texttt{cen}(j')$. Observe that

$$c(\texttt{cen}(j'), j) \leq c(i, j') + c(i, j).$$

Now, $c(i, j') \leq 2\Delta(j')$, and

$$\begin{aligned} c(i, j) \geq c(j, j') - c(i, j') &\overset{\text{Fact (4.1)}}{\geq} 4\Delta(j') - 2\Delta(j') \\ &= 2\Delta(j'). \end{aligned}$$

So, $c(i, j') \leq c(i, j)$, and hence, $c(\texttt{cen}(j'), j) \leq 2c(i, j)$. ∎

(iii) Updating the assignment : We further simplify the assignment of clients to facilities. This will create a new solution $(x^{(2)}, y^{(2)})$, where $y^{(2)}$ will be same as $y^{(1)}$. For a client $j \in \mathcal{D}^{(1)}$, let $\texttt{near}(j)$ be the closest client in the set $\mathcal{D}^{(1)} \setminus \{j\}$. Let $E(j)$ be the ball of radius $c(j, \texttt{near}(j))/2$ around j. Note that by definition, the balls $E(j), j \in \mathcal{D}^{(1)}$, are disjoint. Further, $B(j) \subseteq E(j)$ because the radius of $B(j)$ is $2\Delta(j)$, whereas that of $E(j)$ is at least $2\Delta(j)$ (Fact 4.1). We now modify the solution such that a client j is fractionally assigned to

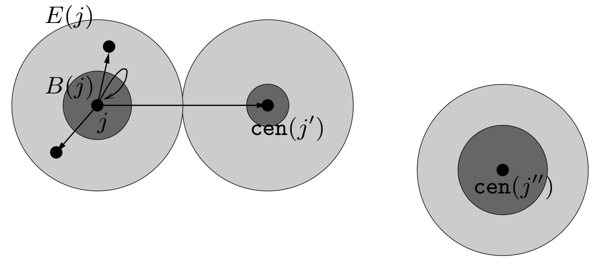

Figure 1: The dark circle around j denotes $B(j)$, and the lighter one denotes $E(j)$. The arrows indicate fractional assignment of j to the facilities. The arrow pointing back to j denotes fractional assignment of j to $\texttt{cen}(j)$.

facilities in either $E(j)$ or $\texttt{cen}(\texttt{near}(j))$. In other words, if $i \notin E(j)$, we set $x^{(2)}(i, j) = 0$ and set

$$x^{(2)}(\texttt{cen}(\texttt{near}(j)), j) = \sum_{i \notin E(j)} x^{(1)}(i, j).$$

Note that this is a feasible solution because $y_{\texttt{cen}(j')} \geq 1/2$ for all j', and at most half of fractional assignment of j goes outside $E(j)$ (because $x^{(1)}(\texttt{cen}(j), j) \geq 1/2$). Thus, a client j is fractionally assigned to $\texttt{cen}(j)$, some facilities in $E(j) \setminus B(j)$, and to at most one facility outside $E(j)$, namely, $\texttt{cen}(\texttt{near}(j))$. We shall often refer to the assignment of j to a facility outside $E(j)$ (right now, this can be only $\texttt{cen}(\texttt{near}(j))$) as a *long range assignment*. See Figure 1 for a pictorial representation of this. Again, it is easy to check the following fact.

Fact 4.3. *For a client j, if $x^{(1)}(i, j) > 0$ for some $i \notin E(j)$, then $c(\texttt{cen}(\texttt{near}(j)), j) \leq 2c(i, j)$. Hence, the cost of $(x^{(2)}, y^{(2)})$ is at most twice that of $(x^{(1)}, y^{(1)})$.*

Proof: Suppose $i \notin E(j)$. Then, $c(i, j) \geq c(\texttt{cen}(\texttt{near}(j)), j)/2$. ∎

(iv) Simplifying the long range assignments We define a new solution $(x^{(3)}, y^{(3)})$ which is initially same as $(x^{(2)}, y^{(2)})$. Consider the following directed graph G, which we call the *long range assignment graph*. The set of vertices V is same as $\mathcal{D}^{(1)}$, and we have a directed arc from j to j' if we have long range assignment from j to $\texttt{cen}(j')$ (this implies that j' is the closest client to j). We change the long range assignments of clients so that this graph has a simple structure. Since the out-degree of each vertex in G is at most one, each component of G is an in-directed tree where the root is either a single vertex or a 2-cycle (it is possible that the root is a larger cycle, but we can always break ties while defining the closest client such that it becomes a 2-cycle). In case

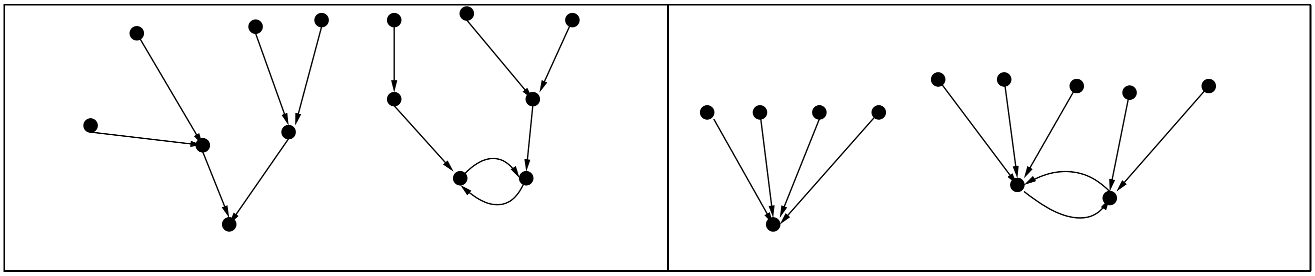

Figure 2: The left figure shows the structure of G – each component is a directed tree with the root being a single node or a 2-cycle. The right figure shows the structure of G after Step (iv). Each component is a directed star with the root being a single node or a 2-cycle.

the root is a 2-cycle, we shall call the vertices in it as *pseudo-roots* (see Figure 2). For a non (pseudo-)root vertex v, we denote its parent as $\texttt{parent}(v)$. We now carry out the following two steps [KKN+11] :

- We traverse each tree in G bottom-up from leaves to (pseudo-)root, and if we encounter a pair of nodes u, v, where $v = \texttt{parent}(u)$, such that v is a not the root or a pseudo-root, and $c(u, v) \leq 2c(v, \texttt{parent}(v))$, then we remove the arc $(v, \texttt{parent}(v))$ and add the arc (v, u) (so these two vertices become a pseudo-root).
- For each component of the graph G, we perform the following changes. If j is a node which is not the root (or one of the pseudo-roots), then we remove the arc $(j, \texttt{parent}(j))$ and add an arc from j to the root (or the closer pseudo-root). Notice that all nodes in this component, except the root (or the pseudo-roots), become leaves. Thus, each component is a star where the root is either a single node or a 2-cycle (right part of Figure 2).

In terms of the actual assignment, if we had an arc (j, j') in G, and the new out-going arc from j is (j, j''), then we set $x^{(3)}(\texttt{cen}(j'), j)$ to 0 and $x^{(3)}(\texttt{cen}(j''), j))$ to $x^{(2)}(\texttt{cen}(j'), j)$.

LEMMA 4.1. *Consider a client j, and suppose j' is the root or one of the pseudo-roots of the component containing j. Then, $c(\texttt{cen}(j'), j) \leq 6c(\texttt{cen}(\texttt{near}(j)), j)$. Thus, the cost of $(x^{(3)}, y^{(3)})$ is at most 6 times that of $(x^{(2)}, y^{(2)})$.*

Proof: Consider the first set of operations above. We only introduce 2-cycles here. Further, if we replace an arc $(j, \texttt{near}(j))$ by (j, j'), then $c(j, j') \leq 2c(j, \texttt{near}(j))$. This proves the lemma when j is a pseudo-root.

Now, consider the graph at the end of the first set of operations. Let j be a vertex which is not a root or pseudo-root vertex. let r be the root (or the closer pseudo-root) of the component containing j. In the path from j to r, the length of the arcs decrease by a factor of at least 2. So, $c(j, r) \leq 2c(j, \texttt{parent}(j))$, where $\texttt{parent}(j)$ is same as $\texttt{near}(j)$. Now, suppose this component has another pseudo-root r'. We argued above that $c(r, r') \leq 2c(r, \texttt{near}(r))$ and so, $c(r, r') \leq 2c(r, j)$. So, $c(j, r') \leq c(j, r) + c(r, r') \leq 3c(j, r) \leq 6c(j, \texttt{parent}(j)) = 6c(j, \texttt{near}(j))$. ■

Summary : We now summarize the properties of the solution $(x^{(3)}, y^{(3)})$.

- We have a set of client $\mathcal{D}^{(1)}$ and a set of facilities $\mathcal{F}^{(1)}$. For each client j, we have a facility $\texttt{cen}(j)$ co-located with j, and $y^{(3)}_{\texttt{cen}(j)} \geq 1/2$.
- The distance between any two clients j and j' is at least $4\max(\Delta(j), \Delta(j')$.
- For each client j, we have two sets : $B(j)$, the ball of radius $2\Delta(j)$ around j; and $E(j)$, the ball of radius $c(j, \texttt{near}(j))/2$ around j, where $\texttt{near}(j)$ is the closest client to j. Clearly, $E(j)$ contains $B(j)$ and for two different clients j and j', $E(j)$ and $E(j')$ are disjoint.
- The only (fractionally) open facility in $B(j)$ is $\texttt{cen}(j)$, and $x^{(3)}(\texttt{cen}(j), j) \geq 1/2$. Further, j may be fractionally assigned to some facilities in $E(j) \setminus B(j)$, and to at most one other facility outside the set $E(j)$. The latter facility is of the form $\texttt{cen}(j')$ for some client j' (by a long range assignment) – in this case, we have an arc (j, j') in G.
- The directed graph G consists of a set of stars, where the root of each star is either a single node or a 2-cycle.

In the initial fractional solution (x, y), a client j is assigned to a facility in its allowable set $\mathcal{A}(j)$ only. However, the solutions constructed by modifying x may violate this assumption. We now show that this condition is still satisfied approximately.

LEMMA 4.2. *Suppose j is a client in $\mathcal{D}^{(1)}$. If $x^{(3)}(i,j) > 0$, then there is a facility $i' \in \mathcal{A}(j)$ for which $c(i,j) \le 24 \cdot c(i',j)$.*

Proof: Fix a client j. In step (i), we do not change any assignments of j. In step (ii), Fact 4.2 shows that if we assign j fractionally to i, $c(i,j) \le 2c(i',j)$, where $i' \in \mathcal{A}(j)$. Fact 4.3 and Lemma 4.1 similarly show that the same condition holds, but the factor worsens to $2 \times 2 \times 6 = 24$. ∎

4.2 A new LP formulation

Consider the instance $\mathcal{I}'$ consisting of clients $\mathcal{D}^{(1)}$ with demands $d_j^{(1)}$ and facilities $\mathcal{F}^{(1)}$. Recall that we have introduced new facilities, $\texttt{cen}(j), j \in D^{(1)}$, and the the facility opening cost of such a facility is same as that of the cheapest facility in $B(j)$. Further, for all clients $j \in \mathcal{D}^{(1)}$, we remove any facility $i \in E(j)$ for which $x^{(3)}(i,j) = 0$. If a client $j \in D^{(1)}$ has an out-neighbor in G, we shall denote it by $\texttt{out}(j)$. For a client j, let $\texttt{comp}(j)$ denote the component of G containing it. For a component C of G, let $\mathcal{R}(C)$ denote the set of (pseudo-)roots of C – if C has just one root, then $\mathcal{R}(C)$ is a singleton set containing this element; otherwise $\mathcal{R}(C)$ consists of the two pseudo-roots of C. We shall denote two pseudo-roots of the same component as *pseudo-root pairs*. For a component C, let $E(\mathcal{R}(C))$ denote $\cup_{j \in \mathcal{R}(C)} E(j)$ – the facilities which belong to $E(j)$, where j is a root or a pseudo-root of C.
We write a new LP relaxation for the instance $\mathcal{I}'$ below – call this relaxation $\texttt{LP}'(\mathcal{I}')$. Note that this LP relaxation was also given by Krishnaswamy et al. [KKN+11].

$$\min \sum_{j \in \mathcal{D}^{(1)}} d_j^{(1)} \left(\sum_{i \in E(j)} c(i,j) y_i \right.$$

$$\left. + c(j, \texttt{out}(j)) \left(1 - \sum_{i \in E(j)} y_i \right) \right) \tag{4.1}$$

$$\sum_{i \in E(j)} y_i \le 1 \quad \forall j \in \mathcal{D}^{(1)} \tag{4.2}$$

$$\sum_{i \in E(\mathcal{R}(C))} y_i \ge 1 \quad \forall \text{components } C \text{ of } G \tag{4.3}$$

$$\sum_i f_i y_i \le B \tag{4.4}$$

$$y_i \ge 0 \quad \forall i$$

Note that for a demand j, the term in the objective function containing $\texttt{out}(j)$ is present only if j has an out-neighbor in G. Note that if a component C of G has only one root j, then constraints (4.2) and (4.3) corresponding to j imply that these should be satisfied with equality. The following claim follows from Facts 4.1, 4.2, 4.3, Lemma 4.1 and the observation that $y^{(3)}$) is a feasible solution to $\texttt{LP}'(\mathcal{I}')$ (with the same cost as the cost of the solution $(x^{(3)}, y^{(3)})$ for the original LP relaxation).

FACT 4.4. *The optimal value of $\texttt{LP}'(\mathcal{I}')$ is at most $24 \cdot \texttt{OPT}(\texttt{LP}(\mathcal{I}))$.*

In this section, we shall prove the following theorem.

THEOREM 4.1. *There is a polynomial time algorithm to find a (integral) solution to the instance $\mathcal{I}'$ whose cost is at most $2700 \cdot \texttt{OPT}(\mathcal{I})$.*

Using the theorem, we get the main result. For a client $j \in \mathcal{D}^{(1)}$, let $D(j)$ denote the set of clients in $\mathcal{D}$ which were merged with j in step (i) of the algorithm described in Section 4.1.

COROLLARY 4.1. *There is a polynomial time algorithm to find a solution to the instance $\mathcal{I}$ of cost at most $2706 \cdot \texttt{OPT}(\mathcal{I})$.*

Proof: Consider the solution $\mathcal{S}'$ to the instance $\mathcal{I}'$ as guaranteed by Theorem 4.1. If $\mathcal{S}'$ opens a facility $\texttt{cen}(j)$ for some $j \in \mathcal{D}^{(1)}$, we open the cheapest facility in $B(j)$. Clearly, this does not change the facility opening cost. This may increase the connection cost by $\sum_{j \in \mathcal{D}^{(1)}} d_j^{(1)} \cdot 2\Delta(j)$. We can express this increase in the cost as

$$2 \sum_{j \in \mathcal{D}^{(1)}} \sum_{j' \in D(j)} d_{j'} \Delta(j) \le 2 \sum_{j' \in \mathcal{D}} d_{j'} \Delta(j') = 2 \cdot \texttt{OPT}(\texttt{LP}(\mathcal{I})).$$

Now, the clients $j' \in D(j)$ also pay the extra distance from j' to j, which is at most $d_{j'} \cdot 4\Delta(j')$. So, we pay an extra $4 \cdot \texttt{OPT}(\texttt{LP}(\mathcal{I}))$. The result now follows from Theorem 4.1. ∎

We now prove Theorem 4.1. First, we state the main technical lemma which shows that a vertex solution to $\texttt{LP}'(\mathcal{I}')$ has nice properties.

LEMMA 4.3. *Any vertex solution $y^\star$ to $\texttt{LP}'(\mathcal{I}')$ has the following property. There is either a client $j^\star$ or a pseudo-root pair $(j^\star, j'^\star)$, such that $y_i^\star$ is either 0 or 1 for all $i \in E(j)$, except perhaps when $j = j^\star$ in the former case and $j = j^\star, j'^\star$ in the latter case. Further, in the latter case the inequality (4.3) for the component containing this pair is satisfied with equality.*

Proof: Fix a vertex solution $y^\star$. We say that a facility i is fractionally open in this solution if $y_i^\star$ is strictly between 0 and 1. We define two sets $\mathcal{A}$ and $\mathcal{B}$. The set $\mathcal{A}$ will contain clients or pseudo-root pairs. A client j gets added to $\mathcal{A}$ if $\sum_{i \in E(j)} y_i^\star = 1$ and $E(j)$ contains fractionally open facilities. Similarly, a pseudo-root pair (j, j') gets added to $\mathcal{A}$ if $\sum_{i \in E(j)} y_i^\star + \sum_{i \in E(j')} y_{i'}^\star = 1$ and both $E(j)$ and $E(j')$ contains fractionally open facilities. Note that in either of the two cases, $E(j)$ (or $E(j) \cup E(j')$) must contain at least two fractionally open facilities. Given a client j (or pseudo-root pair (j, j')) in $\mathcal{A}$, and a small parameter ε, define an operation $T_\varepsilon(j)$ (or $T_\varepsilon((j, j'))$) as follows : there exist two fractionally open facilities in $E(j)$ (or $E(j) \cup E(j')$) – call these i_1 and i_2 respectively. This operation changes $y_{i_1}^\star$ to $y_{i_1}^\star + \varepsilon$ and $y_{i_2}^\star$ to $y_{i_2}^\star - \varepsilon$. Note that this maintains feasibility of all constraints for a small enough but non-zero ε (which could be negative), except perhaps constraint (4.4) which changes by $\varepsilon(f_{i_1} - f_{i_2})$.

The set $\mathcal{B}$ contains those clients j for which $E(j)$ contains a fractionally open facility and j has not been added to $\mathcal{A}$ either as a single element or as part of a pseudo-root pair. So, $\sum_{i \in E(j)} y_i^\star < 1$, and if j is one of the pseudo-roots of $\mathtt{comp}(j)$, then $\sum_{i \in E(\mathcal{R}(\mathtt{comp}(j)))} y_i^\star > 1$. Given a parameter ε' and a client $j \in \mathcal{B}$, define an operation $T_{\varepsilon'}(j)$ as follows : let i be a fractionally open facility in $E(j)$. Then change $y_i^\star$ to $y_i^\star + \varepsilon'$ – it is easy to check that for small enough non-zero ε' (which could be negative), this maintains feasibility of all constraints except perhaps (4.4) which changes by $\varepsilon' f_i$.

Now suppose $\mathcal{A} \cup \mathcal{B}$ has at least two elements, say j and j' (if any these elements is a pseudo-root pair, then the argument is identical). Then perform the operations $T_\varepsilon(j)$ and $T_{\varepsilon'}(j')$ – we can choose ε and ε', where at least one of them is non-zero, such that the left hand side of inequality (4.4) does not change. Also note that switching the signs of ε and ε' also does not violate any of the constraints. This contradicts the fact that we are at a vertex solution. So, $\mathcal{A} \cup \mathcal{B}$ can have cardinality at most 1. This proves the lemma. ■

We now prove two useful claims. The first claim below shows how to handle a client which does not satisfy the statement of Lemma 4.3.

Claim 4.1. *Suppose j is a client in $\mathcal{D}^{(1)}$ such that $x^{(3)}(i, j) > 0$ for some facility i. Then, $d_j^{(1)} \cdot c(i, j) \le 250 \cdot \mathtt{OPT}(\mathcal{I})$.*

Proof: Consider j and i as above. Lemma 4.2 shows that there is a facility $i^\star \in \mathcal{A}(j)$ such that $c(i, j) \le 24 \cdot c(i^\star, j)$. Let $D(j)$ denote the clients in $\mathcal{D}$ which were merged with j. Divide $D(j)$ into two sets – (i) $D^1(j)$: those clients $j' \in D(j)$ for which $c(j, j') \le c(i^\star, j)/2$, and (ii) $D^2(j)$: the remaining clients in $D(j)$. If $j' \in D^2(j)$, then

$$\begin{aligned} c(i, j') &\le c(j, j') + c(i, j) \le c(j, j') + 24 \cdot c(i^\star, j) \\ &\le c(j, j') + 48 \cdot c(j, j') \le 200 \cdot \Delta(j'), \end{aligned}$$

where the last inequality follows because $c(j, j') \le 4\Delta(j')$ (this is the reason why j' was merged with j). Thus, we can pay for the connection cost of clients in $D^2(j)$.

Now, if $j' \in D^1(j)$, then observe that $c(i^\star, j) \le 2(c(i^\star, j) - c(j, j'))$. Hence,

$$\begin{aligned} \sum_{j' \in D^1(j)} d_{j'} \cdot c(i, j) &\le 24 \cdot \sum_{j' \in D^1(j)} d_{j'} c(i^\star, j) \\ &\le 48 \sum_{j' \in D^1(j)} d_{j'} (c(i^\star, j) - c(j, j')) \\ &\le 48 \cdot \mathtt{OPT}(\mathcal{I}), \end{aligned}$$

where the last inequality follows because $c(i^\star, j) \le U_j$. This proves the claim. ■

Claim 4.2. *Suppose (j_1, j_2) is arc in G, and let j_2' be the pseudo-root other than j_2 in this component (if it exists). Let $i \in E(j_1) \cup E(j_2)$. Then $c(i, j_1) \le 9c(j_1, j_2)$.*

Proof: If $i \in E(j_2)$, then $c(i, j_1) \le c(i, j_2) + c(j_1, j_2) \le \frac{c(j_1, j_2)}{2} + c(j_1, j_2) = \frac{3}{2} c(j_1, j_2)$. The case when $i \in E(j_2')$ follows similarly (using Lemma 4.1). ■

We are now ready to prove the main theorem.

Proof of Theorem 4.1: Consider a vertex solution $y^\star$ to $\mathtt{LP}'(\mathcal{I}')$ with properties as stated in Lemma 4.3. We first show which facilities to open. We open all facilities for which $y_i^\star = 1$. Now, consider the fractionally open facilities. Lemma 4.3 implies that all these facilities are either in $E(j^\star)$ for some client $j^\star$ or in $E(j^\star) \cup E(j'^\star)$ for some pseudo-root pair (j, j'). Suppose the first case happens. If $\sum_{i \in E(j^\star)} y_i^\star = 1$, then open the cheapest facility in $E(j)$ – note that this does not increase the budget constraint. Similarly, if the second case happens, then open the cheapest facility in $E(j^\star) \cup E(j^\star)$. It remains to check the assignment cost of clients. First observe that for any component C, constraint (4.3) states that we open at least one facility in $E(\mathcal{R}(C))$.

Now consider a client j. If $y_i^\star = 1$ for some $i \in E(j)$, then we have opened this facility, and in the objective function of $\mathtt{LP}'(\mathcal{I}')$, we also pay for its connection cost. If $y_i^\star = 0$, then the fractional solution $y^\star$ pays $d_j^{(1)} \cdot c(j, \mathtt{out}(j))$. Note that $\mathtt{out}(j)$ is one of the pseudo-roots of $\mathtt{comp}(j)$. We also know we open a facility $i \in E(\mathcal{R}(\mathtt{comp}(j)))$. We assign j to this facility

i. Claim 4.2 implies that we pay at most 9 times what the fractional solution pays for this demand.

Now suppose $E(j)$ contains a fractionally open facility. Lemma 4.3 shows that two options are possible : (i) j is the only client for which $E(j)$ contains a fractionally open facility, or (ii) (j, j') form a pseudo-root pair of a component C and $\sum_{i \in E(\mathcal{R}(C))} y_i^\star = 1$. Further, j and j' are the only clients for which the set $E(j)$ contains a fractionally open facility.

Consider case (i) first. First assume that j is the root of $\texttt{comp}(j)$. Then, we have opened a facility i in $E(j)$. Lemma 4.3 shows that its connection cost is at most $250 \cdot \texttt{OPT}(\mathcal{I})$. Now suppose either j is a leaf or one of the pseudo-roots. Since we always open a facility in $E(\mathcal{R}(\texttt{comp}(j)))$, Lemma 4.3 and Claim 4.2 show that the assignment cost of j is at most $9 \cdot 250 \cdot \texttt{OPT}(\mathcal{I}) = 2250 \cdot \texttt{OPT}(\mathcal{I})$.

Now, suppose case (ii) as above happens. We need to worry about the connection cost of j and j'. Since $\sum_{i \in E(j)} y_i^\star + \sum_{i \in E(j')} y_i^\star = 1$, one of the two summands is at most $1/2$ – suppose it is j. Then the LP objective function pays at least $1/2 \cdot d_j^{(1)} c(j, j')$. We know that we have opened a facility $i \in E(j) \cup E(j')$. Claim 4.2 shows that the assignment cost of j is at most $9 \cdot d_j^{(1)} c(j, j')$, and hence, at most 18 times the contribution from the LP objective function. For j', Lemma 4.3 and Claim 4.2 show that the assignment cost of j' is at most $9 \cdot 250 \cdot \texttt{OPT}(\mathcal{I}) = 2250 \cdot \texttt{OPT}(\mathcal{I})$.

So, the total assignment cost of all the clients is at most $18 \cdot \texttt{LP}'(\mathcal{I}') + 2250\texttt{OPT}(\mathcal{I})$, which is at most $2700 \cdot \texttt{OPT}(\mathcal{I})$ (using Fact 4.4). ■

5 Acknowledgments

The author would like to thank the anonymous referee whose suggestions helped in simplifying the algorithm.

References

[AGK+01] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *ACM Symposium on Theory of Computing*, pages 21–29, 2001.

[CG99] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *IEEE Foundations of Computer Science*, pages 378–388, 1999.

[CG05] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4):803–824, 2005.

[CGTS02] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.

[HKK10] MohammadTaghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Budgeted red-blue median and its generalizations. In *ESA (1)*, pages 314–325, 2010.

[JV01] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[KKN+11] Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. In *ACM/SIAM Symposium on Discrete Algorithms*, pages 1117–1130, 2011.

[KPR98] Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. In *ACM/SIAM Symposium on Discrete Algorithms*, pages 1–10, 1998.

[LMW98] R. F. Love, J. G. Morris, and G. O. Wesolowsky. *Facilities Location: Models and Methods.* North-Holland, New York, NY, 1998.

[LV92] Jyh-Han Lin and Jeffrey Scott Vitter. epsilon-approximations with minimum packing constraint violation (extended abstract). In *ACM Symposium on Theory of Computing*, pages 771–782, 1992.

[Mes07] Julián Mestre. Lagrangian relaxation and partial cover. *CoRR*, abs/0712.3936, 2007.

[MF90] P. Mirchandani and R. Francis. *Discrete Location Theory.* Wiley, New York, NY, 1990.