Facility Location with Service Installation Costs

(Extended Abstract)

David B. Shmoys* Chaitanya Swamy* Retsef Levi†

Abstract

We consider a generalization of the uncapacitated facility location problem which we call *Facility Location with Service Installation Costs*. We are given a set of facilities, $\mathcal{F}$, a set of demands or clients $\mathcal{D}$, and a set of services $\mathcal{S}$. Each facility i has a *facility opening cost* f_i, and we have a *service installation cost* of f_i^l for every facility-service pair (i, l). Each client j in $\mathcal{D}$ requests a specific service $g(j) \in \mathcal{S}$ and the cost of assigning a client j to facility i is given by c_{ij}. We want to open a set of facilities, install services at the open facilities, and assign each client j to an open facility at which service $g(j)$ is installed, so as to minimize the sum of the facility opening costs, the service installation costs and the client assignment costs.

Our main result is a primal-dual 6-approximation algorithm under the assumption that there is an ordering on the facilities such that if i comes before i' in this ordering then for *every* service type l, $f_i^l \le f_{i'}^l$. This includes (as special cases) the settings where the service installation cost f_i^l depends only on the service type l, or depends only on the location i. With arbitrary service installation costs, the problem becomes as hard as the set-cover problem. Our algorithm extends the algorithm of Jain & Vazirani [9] in a novel way. If the service installation cost depends only on the service type and not on the location, we give an LP rounding algorithm that attains an improved approximation ratio of 2.391. The algorithm combines both clustered randomized rounding [6] and the filtering based technique of [10, 14]. We also consider the k-median version of the problem where there is an additional requirement that at most k facilities may be opened. We use our primal-dual algorithm to give a constant-factor approximation for this problem when the service installation cost depends only on the service type.

1 Introduction

Facility location problems have been widely studied in the Operations Research community (see for e.g. [12]). In its simplest version, uncapacitated facility location (UFL), we are given a set of facilities, $\mathcal{F}$, and a set of demands or clients $\mathcal{D}$. Each facility i has a *facility opening cost* f_i and the cost of assigning a client j to facility i is given by c_{ij}. We want to *open* some facilities from the set $\mathcal{F}$ and assign each demand to an open facility. The goal is to minimize the sum of the facility opening costs and the client assignment costs. This problem has a wide range of applications. For example, a company might want to open its warehouses at some locations so that its total cost of opening warehouses and servicing customers is minimized.

In various applications, the clients are differentiated according to the kind of service they require and to satisfy the service requirement of a client we have to assign it to a facility that can provide the service required by the client. For example, in the warehouse location problem above, the customers may be retail stores that request different kinds of supplies. A warehouse may store different kinds of supplies. To satisfy a customer we have to assign it to a warehouse that holds inventory of the type requested by the customer. We model such a setting by saying that in addition to facilities and clients, we have a set of services $\mathcal{S}$. Each client j in $\mathcal{D}$ requests a specific service $g(j) \in \mathcal{S}$. To satisfy client j we have to assign it to an open facility on which service $g(j)$ is installed. Further, if we install service l on an open facility i we incur a *service installation cost* of f_i^l. We want to open a set of facilities, install services at the open facilities, and assign each client j to an open facility i such that service $g(j)$ is installed at i. The cost of a solution is the sum of the facility opening costs, the service installation costs and the client assignment costs, and the goal is to find a solution with minimum total cost. We call this problem, *Facility Location with Service Installation Costs*. In the warehouse location problem, the service installation cost corresponds to the initial cost of setting up the warehouse to store the particular kind of inventory. The notion of service-dependent fixed costs is also used in inventory problems where one incurs a joint setup cost to start a new order and an item-dependent fixed cost to order a specific item, so one needs to coordinate the placement of item orders; see [1] for a survey. We assume throughout that the assignment costs c_{ij} form a metric.

*{`shmoys,swamy`}`@cs.cornell.edu`. Dept. of Computer Science, Cornell University, Ithaca, NY 14853. Research supported partially by NSF grant CCR-9912422.

†`rl227@cornell.edu`. School of ORIE, Cornell University, Ithaca, NY 14853. Research supported partially by a grant from Motorola and NSF grant CCR-9912422.

Applications and Related Work. Facility location with service installation costs can be used to model a data management/caching problem. Here we are given a set of locations in a network at which caches may be built and a set of processes located at the nodes of the network that request data items. Each process requests a specific data item. To satisfy the request we must assign the process to a cache that stores the requested data item, incurring an access cost proportional to the distance between the process site and the cache location. Building a cache at a location incurs a location dependent cost and storing a data item in a cache at a particular location incurs a cost that depends on the data item and the location. The goal is to build caches, store data items in the caches and assign each process to a cache containing the data item requested by the process, so as to minimize the total cost of building caches, storing data items and the access cost of requests. This is exactly the facility location problem with service installation costs where the caches are facilities, the processes are clients and the data items correspond to services. Baev & Rajaraman [3] considered a closely related problem called the *data placement* problem and gave a 20.5-approximation algorithm. Here caches of fixed capacity are already built at certain locations and the goal is to find a placement of data items to caches respecting the cache capacities that minimizes the sum of the access costs and the cost of storing data items. The ratio has recently been improved by Swamy [16].

Facility location with service installation costs is a generalization of UFL — if there is just one service type then this is simply the uncapacitated facility location problem. There is a large body of literature that deals with designing approximation algorithms for metric UFL and we sample only a few results below; see [13] for a survey of this and earlier work. Shmoys, Tardos & Aardal [14] gave the first constant-factor approximation algorithm for this problem using the *filtering* technique of Lin & Vitter [10] to round the optimal solution of a linear program. Chudak & Shmoys [5, 6] gave an LP rounding based $\left(1+\frac{2}{e}\right)$-approximation algorithm. They combined randomized rounding and the decomposition results of [14] to get a variant that might be called *clustered randomized rounding*. Sviridenko [15] improved the ratio to 1.58. Jain & Vazirani [9] gave a combinatorial *primal-dual* 3-approximation algorithm where the LP is used only in the analysis. The current best ratio for UFL is 1.52 [11] obtained by building upon a dual-fitting based greedy algorithm of Jain, Mahdian, Markakis, Saberi & Vazirani [8].

Our Results. Our main result is a primal-dual 6-approximation algorithm for the facility location problem with service installation costs under the assumption that there is an ordering on the facilities such that if i comes before i' in this ordering then for *every* service type l, $f_i^l \leq f_{i'}^l$. This is reasonable in many settings; for example, one expects the inventory setup cost of a warehouse in New York city to be less than the inventory setup cost in a remote town like Ithaca regardless of the kind of inventory. As special cases this includes the cases where the service installation cost f_i^l depends only on the service type l, or depends only on the location i. In the former setting where the service installation cost depends only on the service type we give an algorithm based on LP rounding that attains a much improved approximation ratio of 2.391. We show that with arbitrary service installation costs the problem becomes as hard as the set-cover problem. Combined with the result of Feige [7], this shows that no polynomial-time algorithm with a ratio of $(1-\epsilon)\ln|\mathcal{D}|$ exists for this problem in the general case unless NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$. We also consider the k-median version of the problem where there is an additional requirement that at most k facilities may be opened. We use our primal-dual algorithm to give a constant-factor approximation for this problem when the service installation cost depends only on the service type.

Our Techniques. Facility location with service installation costs is a generalization of UFL. It differs however from traditional multi-level extensions of facility location where we assume that a demand can be assigned to any facility in any level. In our problem demand j may only be assigned to a level 1 "facility" $(i, g(j))$ and then to facility i in level 2. Moreover, existing techniques for UFL and multi-level facility location do not readily generalize. If there were no facility opening costs we could decouple the problem into several UFL instances, one for each service type, and solve each one separately. With facility opening costs this approach fares badly since we may end up opening a lot of facilities and spend too much on the facility opening costs. All known algorithms for UFL rely on the fact, either in the design or the analysis, that a client j can be moved from a facility i to another nearby facility i' without increasing its assignment cost by much, and leaving the facility opening cost unchanged. In our problem, reassigning j to i' may now require us to install service $g(j)$ on i' causing us to pay the installation cost $f_{i'}^{g(j)}$ which could be large. Technically the hard part is to find a way to reassign clients to nearby facilities so that we do not pay too much to install services at the new locations. With arbitrary service installation costs such a reassignment need not be possible since we can encode the constraint that a client may only be assigned to a specific set of facilities, making the problem set-cover hard.

We build upon the primal-dual algorithm of [9] in a novel way and give a 6-approximation algorithm under the assumption that the facilities are ordered so that if i comes before i' then $f_i^l \leq f_{i'}^l$ for every service l. At a high level, the idea is to consider an integer programming formulation of the problem and the dual of its linear programming relaxation, and construct simultaneously an integer primal solution and a dual solution. Each client j has a dual variable α_j which can be interpreted intuitively as the payment that j is willing to make to get itself assigned to an open facility.

The Jain-Vazirani (JV) algorithm [9] for UFL works in two phases. In phase I we grow each dual variable α_j uniformly and gradually build a primal feasible solution. Once α_j becomes equal to c_{ij} for some facility i, j starts paying toward the facility opening cost of i. When the total contribution to i from the various clients equals f_i, we declare i to be tentatively open and assign all the unassigned clients contributing toward i to i. Phase I ends when each client is assigned to a tentatively open facility. At this point a client could be contributing towards multiple tentatively open facilities. We call a set of facilities *independent* if each client contributes towards at most one facility in the set. In phase II we select a *maximal independent subset* of tentatively open facilities and open these. The analysis shows that if the facility i to which a client j was assigned in phase I is not opened, then there is a "nearby" open facility i' to which j can be reassigned.

Our algorithm also proceeds in phases. During phase I we tentatively open some facilities, tentatively install service l on some facilities for each service type l, and assign each client j to a tentatively open facility on which service $g(j)$ is tentatively installed. Phase II is more involved. We have to select a set of facilities to open and install services in such a way that we can, (1) pay for installing the services, and (2) ensure that if a client j has to be reassigned there is a nearby open facility on which service $g(j)$ is installed. We show that we can achieve properties (1) and (2) if we look at the facilities in a particular order and pick a maximal independent subset greedily. This gives us a 6-approximation algorithm.

Our primal-dual algorithm exploits the property that in the JV algorithm *any maximal independent set of tentatively open locations may be picked.* Although this is a well-known fact, to our knowledge this has been used in only a couple of applications previously. Bartal, Charikar and Raz [4] consider a clustering problem and use the JV algorithm to solve a relaxation of the problem by picking an appropriate maximal independent set. Archer, Rajagopalan and Shmoys [2] pick a *maximum* independent set and use this to prove a bound on the integrality gap of the k-median LP.

When the service installation cost depends only on the service type, we give an LP-rounding algorithm that combines clustered randomized rounding [6] and the filtering based technique of [10, 14]. A feature of the algorithm is that we bound distances c_{ij} using both the α_j bound due to complementary slackness and the bound obtained by filtering. This gives a better performance guarantee than that obtained by using either of the two bounds separately. Sviridenko [15] also used the two bounds in conjunction to improve the approximation ratio for UFL from $\left(1+\frac{2}{e}\right)$ to 1.58.

2 A Linear Program

We can formulate the problem as an integer program and relax the integrality constraints to get a linear program. We use i to index the facilities in $\mathcal{F}$, j to index the clients in $\mathcal{D}$ and l to index the services in $\mathcal{S}$.

$$\begin{aligned}
\min \quad & \sum_i f_i y_i + \sum_i \sum_l f_i^l y_i^l + \sum_j \sum_i c_{ij} x_{ij} && \text{(P)} \\
\text{s.t.} \quad & \sum_i x_{ij} \geq 1 && \forall j \\
& x_{ij} \leq y_i^{g(j)} && \forall i, j \\
& x_{ij} \leq y_i && \forall i, j \\
& x_{ij}, y_i, y_i^l \geq 0 && \forall i, j, l.
\end{aligned}$$

Variable y_i indicates if facility i is open, y_i^l indicates if service type l is installed at i, and x_{ij} indicates if client j is connected to facility i. The first constraint states that each client must be assigned to a facility, the second and the third constraints say that if client j is assigned to facility i, then service $g(j)$ must be installed on i and i must be open. An integral solution corresponds exactly to a solution to our problem. Let G_l be the set of clients requesting service l. The dual program is,

$$\begin{aligned}
\max \quad & \sum_j \alpha_j && \text{(D)} \\
\text{s.t.} \quad & \alpha_j \leq c_{ij} + \beta_{ij} + \theta_{ij} && \forall i, j \qquad (2.1) \\
& \sum_{j \in G_l} \theta_{ij} \leq f_i^l && \forall i, l \\
& \sum_j \beta_{ij} \leq f_i && \forall i \qquad (2.2) \\
& \alpha_j, \beta_{ij}, \theta_{ij} \geq 0 && \forall i, j.
\end{aligned}$$

Intuitively α_j is the *budget* that j is willing to spend to get itself assigned to an open facility. Constraint (2.1) says that a part of this goes towards paying for the assignment cost c_{ij}. The rest gets divided into a payment for the service installation cost θ_{ij}, and a payment for the facility opening cost β_{ij}.

3 A Primal-Dual Algorithm

We consider instances of the problem where there is an ordering on the facilities in $\mathcal{F}$ such that if i comes before i' in this ordering then for *every* service type l, $f_i^l \leq f_{i'}^l$. Equivalently this means that for any two locations i, i', the vectors $\left(f_i^l\right)^{\mathrm{T}}_{l=1\ldots|\mathcal{S}|}$ and $\left(f_{i'}^l\right)^{\mathrm{T}}_{l=1\ldots|\mathcal{S}|}$ are comparable. Let $\mathcal{O}$ denote this total ordering on the facilities. We say that $i \leq i'$ if i comes before i' in the ordering $\mathcal{O}$.

The algorithm is strongly motivated by the primal-dual algorithm of Jain and Vazirani for the traditional uncapacitated facility location algorithm. In our algorithm, we first construct a feasible dual solution, and then use this dual solution to extract a feasible (integer) primal solution. As has been the norm, our algorithm is a dual ascent algorithm, so all dual variables are only increased throughout the execution of the algorithm.

We next describe the algorithm. There is a notion of time around which the algorithm is specified. We start at time $t = 0$, all dual variables are initialized to 0, each demand j is said to be *unfrozen*, and all facilities are closed. At first, the variables that are increased are the α_js; more precisely, for any unfrozen demand j, α_j is always equal to the time t. We say that *demand* j is *tight* with facility i, or has reached i, if $\alpha_j \geq c_{ij}$. As time increases, we will *freeze* demand points, *tentatively open* facilities, and *tentatively install services* at facilities. We increase the α_j of each demand j until one of the following events happens:

1. Suppose that demand j becomes tight with facility i. If service $g(j)$ is not tentatively installed at i, then we start increasing θ_{ij} at the same rate as α_j; that is, if $\alpha_j = t$, then $\theta_{ij} = t - c_{ij}$. If service $g(j)$ is tentatively installed, but i is not tentatively open, we instead increase β_{ij} at the same rate as α_j; that is, if $\alpha_j = t$, then θ_{ij} remains 0, but $\beta_{ij} = t - c_{ij}$. Finally, if service $g(j)$ is tentatively installed, and i is tentatively open, we freeze demand point j (and no longer increase α_j).

2. Suppose that for a facility i and a service type l, we get that $\sum_{j \in G_l} \theta_{ij} = f_i^l$: in this case, we tentatively install service l at i. If i is also tentatively open, then we freeze each demand $j \in G_l$ that is tight with i (and no longer increase α_j). If i is not yet tentatively open, then for each demand $j \in G_l$ that is tight with i, we no longer increase θ_{ij}, but instead start increasing β_{ij} at the same rate as α_j.

3. Suppose that for a facility i, $\sum_j \beta_{ij} = f_i$: in this case, we tentatively open i. For each demand j, we do not increase β_{ij} from now on. If demand j is tight with i and service $g(j)$ is tentatively installed at i, we freeze j (and no longer increase α_j).

We only raise the $\alpha_j, \beta_{ij}, \theta_{ij}$ of unfrozen demands. Frozen demands do not participate in any events. We continue this process until all demands become frozen. Let (α, β, θ) denote the final dual solution obtained by the above process. Observe that if i is the facility that caused j to freeze, then service $g(j)$ must be tentatively installed at i, and i must be tentatively open.

We now specify which facilities to open, how to install services on facilities, and how to assign demands to facilities. Let F be the set of tentatively open facilities, and let $F_l \subseteq F$ be the set of tentatively open facilities on which service l is tentatively installed. For facility $i \in F$, let t_i be the time at which i became tentatively open. If $i \in F_l$, let t_{il} be the time at which service l was tentatively installed at i.

Opening facilities. We open a subset of facilities from F. We say that $i, i' \in F$ are *dependent* if there is a demand j such that both β_{ij} and $\beta_{i'j}$ are positive. We consider the facilities in F in the order given by $\mathcal{O}$ and pick a maximal independent set of facilities, $F' \subseteq F$. We open the facilities in F'.

Installing services. Consider service type l and the set of facilities F_l. We say that facilities $i, i' \in F_l$ are service-l-dependent if there exists some demand j in G_l such that both θ_{ij} and $\theta_{i'j}$ are positive. We pick a maximal independent subset F_l' by looking at facilities in F_l in a particular order: first we consider facilities in $F_l \cap F'$ in increasing order of t_{il}, and then facilities in $F_l - F'$ in increasing order of t_i.

We first install service l on all facilities in $F_l' \cap F'$. Furthermore, for each $i \in F_l' - F'$, we pick a facility $i' \in F'$ such that i' and i are dependent and $i' \leq i$ (in the ordering given by $\mathcal{O}$), and install service l on facility i'. We say that i' is the neighbor of i and denote it by $\mathsf{nbr}(i)$. Note that $\mathsf{nbr}(.)$ depends only on i and not on the service type l: if $i \notin F'$, then we can choose a single facility $i' \in F'$ such that $i' \leq i$ regardless of the service type l.

Assigning demands. We assign each client j to the nearest open facility at which service $g(j)$ is installed.

3.1 Analysis.

We now bound the performance of our algorithm. The following lemma just says what it means for a demand j to get frozen.

LEMMA 3.1. *Let i be the facility that causes a demand*

j to freeze. Then, i is tentatively open, service $g(j)$ is tentatively installed at i, and $\alpha_j = \max(c_{ij}, t_i, t_{ig(j)})$.

We start by bounding the cost incurred in opening facilities, and installing services. Let D' be the subset of demands $\{j : \exists i \in F' \text{ s.t. } \beta_{ij} > 0\}$.

LEMMA 3.2. *The cost of opening facilities is at most $\sum_{j \in D'} \alpha_j$. Furthermore, the cost of installing services is at most $\sum_j \alpha_j$.*

Proof. For each facility i that is tentatively opened, we have that $f_i = \sum_{j \in \mathcal{D}} \beta_{ij}$. By the construction of F', we know that for each demand j, there is at most one $i \in F'$ such that $\beta_{ij} > 0$. Summing over all facilities, and using this fact, we see that

$$\sum_{i \in F'} f_i = \sum_{i \in F'} \sum_{j \in \mathcal{D}} \beta_{ij} = \sum_{j \in D'} \sum_{i \in F'} \beta_{ij} \leq \sum_{j \in D'} \alpha_j.$$

By the definition of $\mathsf{nbr}(i)$, we know that for any service type l, $f^l_{\mathsf{nbr}(i)} \leq f^l_i$. For each $i \in F'_l$, we install service l either at $i \in F'_l$ or at $\mathsf{nbr}(i)$, and so we can upper bound the total cost of installing services of type l by $\sum_{i \in F'_l} f^l_i$. Since each service l is tentatively installed only when $f^l_i = \sum_{j \in G_l} \theta_{ij}$, we have that $\sum_{i \in F'_l} f^l_i = \sum_{j \in G_l} \sum_{i \in F'_l} \theta_{ij}$. The notion of service-$l$-dependence insures that for each demand $j \in G_l$, there is at most one facility $i \in F'_l$ for which θ_{ij} is positive. We obtain that the total cost of installing service l is at most $\sum_{j \in G_l} \alpha_j$, which immediately implies the lemma.

We next bound the assignment cost incurred by the solution computed. The following facts, which follow directly from the construction of the algorithm, will be useful in this analysis.

FACT 3.1. *Suppose that β_{ik} is positive. Then it follows that $c_{ik} \leq \alpha_k - \beta_{ik}$ and $\alpha_k \leq t_i$.*

FACT 3.2. *Suppose that θ_{ik} is positive. Then $c_{ik} \leq \alpha_k - \theta_{ik}$ and $c_{ik} < t_{ig(k)}$. If $\beta_{ik} = 0$ then $\alpha_k \leq t_{ig(k)}$.*

For example, we use these in deriving the following bounds.

CLAIM 3.1. *If i and i' are dependent facilities in F, then $c_{ii'} < 2\min(t_i, t_{i'})$.*

Proof. Let k be a client such that β_{ik} and $\beta_{i'k}$ are positive. Applying Fact 3.1 for both of these, and applying the triangle inequality, we get that $c_{ii'} < 2\alpha_k \leq 2\min(t_i, t_{i'})$.

CLAIM 3.2. *Let $i, i' \in F_l$ be service-l-dependent due to demand $k \in G_l$. Then $c_{ii'} < 2\max(t_{il}, t_{i'l})$ and both c_{ik} and $c_{i'k}$ are less than α_k.*

Proof. From the dependence of i and i', it follows that θ_{ik} and $\theta_{i'k}$ are positive. Applying Fact 3.2 for both of these, and using the triangle inequality, we get that $c_{ii'} < 2\max(t_{il}, t_{i'l})$, $c_{ik} < \alpha_k$ and $c_{i'k} < \alpha_k$.

LEMMA 3.3. *If $j \in D'$, then the assignment cost incurred for j is at most $3\alpha_j$; if $j \notin D'$, then the assignment cost incurred for j is at most $5\alpha_j$.*

Proof. We will show that there always exists some open facility with service $g(j)$ installed that is no further from j than the claimed bound (and hence the closest one, to which j is assigned, is no further away).

Consider $j \in D'$ with $g(j) = l$. Let i be the unique facility in F' for which β_{ij} is positive. If $i \in F'_l$, then we have installed service l at i, and $c_{ij} \leq \alpha_j - \beta_{ij}$. Otherwise, i' and i are service-l-dependent for some i' in $F' \cap F'_l$ with $t_{i'l} \leq t_{il}$. So, by Claim 3.2, $c_{ii'} < 2\max(t_{il}, t_{i'l}) = 2t_{il}$. Since $\beta_{ij} > 0$, it follows that $t_{il} \leq \alpha_j - \beta_{ij}$. So by the triangle inequality, $c_{i'j} \leq 3\alpha_j$.

Now consider a demand $j \notin D'$, and again let $g(j) = l$. Let i be the facility that caused j to freeze, and so $\alpha_j = \max(c_{ij}, t_i, t_{il})$. If $i \in F' \cap F'_l$, then service l is installed at i, and $c_{ij} \leq \alpha_j$. Suppose that $i \in F'_l - F'$, and let $i' = \mathsf{nbr}(i)$ (an open facility at which service l is installed, see Fig. 1a). By Claim 3.1, $c_{ii'} \leq 2t_i \implies c_{i'j} \leq 3\alpha_j$.

Next suppose that $i \notin F'_l$. Since $i \in F_l$, there must exist $i' \in F'_l$ such that i was not picked in F'_l because i and i' are service-l-dependent due to a client k. If i' is also in F' (Fig. 1b), then service l is installed there. Applying Claim 3.2, we obtain that both $c_{i'k}$ and c_{ik} are less than $\alpha_k \leq \max(t_i, t_{il}) \leq \alpha_j$, and hence $c_{i'j} \leq 3\alpha_j$. However, if $i' \notin F'$ (so $i \notin F'$), then let $i'' = \mathsf{nbr}(i')$; service l is installed at i'' (Fig. 1c). Since i' and i service-l-dependent, we have that $t_{i'} \leq t_i$, $c_{i'i''} \leq 2\min(t_{i'}, t_{i''})$ (Claim 3.1), and $c_{i'j} \leq 3\alpha_j$ as above, which implies that $c_{i''j} \leq 5\alpha_j$. This completes the proof.

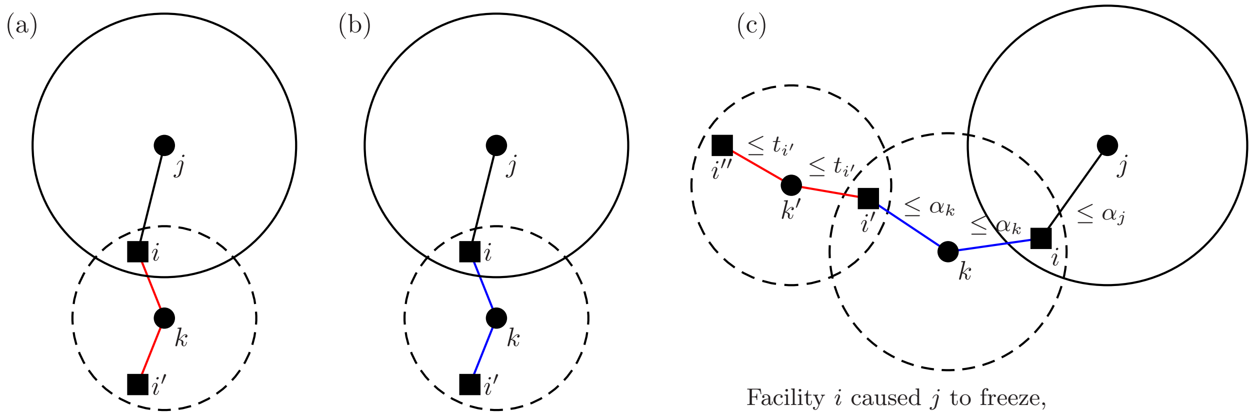
THEOREM 3.1. *The above algorithm returns a solution of cost at most $6\sum_j \alpha_j \leq 6 \cdot OPT$.*

Proof. Follows from Lemma 3.2 and Lemma 3.3.

4 An LP-Rounding Algorithm

We now give an algorithm for the special case where $f^l_i = f^l$, i.e., the installation cost depends only on the service l and not on the location at which it is installed. Note that this case is handled by the primal-dual algorithm above. Here we adapt the rounding procedure of [6] to give deterministic and randomized approximation algorithms achieving ratios of 6 and 2.391 respectively.

Let (x, y) and (α, β, θ) be the optimal solutions to (P) and (D) respectively. We can ensure that for



$i \in F'_l - F'$, $i' = \mathsf{nbr}(i)$

$i \notin F'_l$, i', i are service-l-dependent

Facility i caused j to freeze, i and i' are service-l-dependent, $i'' = \mathsf{nbr}(i')$

Figure 1: Bounding the assignment cost of j. (a), (b) Different 3-hop cases, and (c) the 5-hop case.

every i, j and l, $x_{ij} = 0$ or $x_{ij} = y_i^{g(j)}$ and $y_i^l = 0$ or $y_i^l = y_i$. We will round the fractional solution (x, y) to an integer solution losing a factor of at most 6. Let $F_j = \{i : x_{ij} > 0\}$. We describe the algorithm briefly.

A1. First, for every service type l, we consider the clients in G_l and cluster the facilities on which service l is installed around some cluster centers: pick $j \in G_l$ with smallest α_j value and form a cluster around j consisting of the facilities in F_j. We remove every client $k \in G_l$ (including j) that is assigned (fractionally) to some facility in the cluster created, and recurse on the remaining set of clients until no client in G_l is left. Let D_l be the set of cluster centers.

A2. Let $D = \bigcup_l D_l$. We cannot open a facility in every cluster since different clusters could share the same fractional facility weight (y_i) if the cluster centers request different services. Say that $j, k \in D$ are dependent if $F_j \cap F_k \neq \phi$. Note that this can only happen if j and k request different services. We consider clients in D in order of increasing α_j and pick a maximal independent subset D'. For each client $j \in D'$, we open the facility in F_j with smallest f_i and install service $g(j)$ on it. Further for every $k \in D - D'$, there is some $j \in D'$ with $\alpha_j \leq \alpha_k$ such that j and k are dependent. We pick some such j and install service $g(k)$ on the facility opened from F_j. Call j the neighbor of k and denote it $\mathsf{nbr}(k)$.

A3. We assign demand j to a facility as follows: (i) if $j \in D'$ it is assigned to the facility opened from F_j, (ii) if $j \in D - D'$, then $\mathsf{nbr}(j) = k \in D'$ and j is assigned to the facility opened from F_k, and (iii) if $j \notin D$, there is some $k \in D_{g(j)} \subseteq D$ such that $\alpha_k \leq \alpha_j$ and j was removed from G_l in step A1 because a cluster was created around k; we assign j to the same facility as k. Note that this is a feasible assignment of demands to facilities.

Lemma 4.1. *The facility opening cost is at most* $\sum_i f_i y_i$.

Proof. The clusters corresponding to clients in D' are disjoint and we open the cheapest facility in each cluster — the lemma follows.

Lemma 4.2. *The cost of installing services is at most* $\sum_{i,l} f_i^l y_i^l$.

Proof. The cost of installing a service is independent of the location at which it is installed, and for any service type l, we install service on at most one (new) location per cluster center in D_l. So the total cost of installing services is at most $\sum_l \sum_{j \in D_l} f^l = \sum_l \sum_{j \in D_l} f^l \sum_{i \in F_j} y_i \leq \sum_{i,l} f_i^l y_i^l$.

Lemma 4.3. *The assignment cost of client j is at most* $5\alpha_j$.

Proof. If $j \in D'$, we assign j to a facility $i \in F_j$, and $c_{ij} \leq \alpha_j$ by complementary slackness. If $j \in D - D'$ and $\mathsf{nbr}(j) = k \in D'$, we assign j to the facility i opened from F_k. Since $\alpha_k \leq \alpha_j$ and $c_{jk} \leq 2\alpha_j$, $c_{ij} \leq 3\alpha_j$. If $j \notin D$, j is assigned to the same facility as k' where $k' \in D_{g(j)} \subseteq D$ and j was removed from $G_{g(j)}$ in step A1 because a cluster was created around k'. So $\alpha_{k'} \leq \alpha_j$ and $c_{jk'} \leq 2\alpha_j$. From above, k' is assigned to a facility i with $c_{ik'} \leq 3\alpha_{k'}$, so $c_{ij} \leq 5\alpha_j$.

Thus we have proved the following theorem.

Theorem 4.1. *The cost of the solution returned is at most* $6 \cdot OPT$.

4.1 Improvement using randomization. We give a rounding procedure that combines clustered randomized rounding [6] and the filtering based technique of [10, 14]. We define some notation first. Let $0 < \gamma < 1$ be a parameter that we will set later and $r = \frac{1}{\gamma}$. Sort the facilities in F_j by increasing c_{ij}. Let i' be the first facility in this ordering such that $\sum_{i \in F_j : c_{ij} \le c_{i'j}} x_{ij} \ge \gamma$. Let N_j be the subset of F_j consisting of all facilities (including i') that come before i' in this ordering. Define $C_j(\gamma) = c_{i'j}$ and $\bar{C}_j = \sum_i c_{ij} x_{ij}$. To simplify things we assume that each $y_i \le \gamma$ and for any j, $\sum_{i \in N_j} y_i$ is exactly γ. If some $y_i > \gamma$, then we can create at most $\lceil 1/\gamma \rceil$ copies of i and set $y_i \le \gamma$ for each of the copies so that $\sum_{\text{copies } i'} y_{i'} = y_i$ (setting the variables $x_{i'j}, y^l_{i'}$ accordingly). Similarly if $\sum_{i \in N_j} y_i > \gamma$, we can take the facility i' in N_j that comes last in the ordering in F_j and split it into two copies i'_1 and i'_2 setting $y_{i'_2} = \sum_{i \in N_j} y_i - \gamma$, $y_{i'_1} = y_{i'} - y_{i'_2}$ (and the other variables accordingly). We include only i'_1 in N_j thus ensuring that $\sum_{i \in N_j} y_i = \gamma$. The cost of the fractional solution remains unchanged by these transformations and any solution to the modified instance gives a solution to the original instance of no greater cost.

R1. This is the same as step A1 except that we choose $j \in G_l$ with smallest $2\alpha_j + C_j(\gamma) + \bar{C}_j$ as the cluster center. This gives us a set of cluster centers D_l for each service type l.

R2. We prune the set $D = \bigcup_l D_l$ as in step A2 but modify the notion of dependency to say that $j, k \in D$ are dependent if $N_j \cap N_k \ne \phi$, and consider the clients in D in increasing order of $C_j(\gamma) + \bar{C}_j$. For $k \in D - D'$ we define $\mathsf{nbr}(k)$ as before. We call the facilities in N_j for clients $j \in D'$ *central facilities*, and the rest as *non-central facilities*.

R3. For every client $j \in D'$ we randomly open exactly one facility in N_j by choosing facility i with probability $y_i / \sum_{i \in N_j} y_i = r \cdot y_i$. This facility now serves as a *backup facility* for all the clients that would get assigned to this facility in step A3 of the deterministic algorithm.

R4. Independent of step R3, each non-central facility i is opened independently with probability $r \cdot y_i$.

R5. For *any* facility i, be it a central or a non-central facility, if i is opened (in R3 or R4), we install on it all services that are installed on it in the fractional solution, i.e., all l such that $y^l_i > 0$.

R6. For every client $j \in D - D'$, if no facility from F_j is open, we install service $g(j)$ on the facility opened in R3 from $N_{\mathsf{nbr}(j)}$.

R7. We assign demand j to the nearest open facility at which service $g(j)$ is installed.

Lemmas 4.1 and 4.2 are modified to the following.

Lemma 4.4. *The expected cost of opening facilities is $r \cdot \sum_i f_i y_i$. The expected cost of installing services is at most $\left(r + \frac{1}{e^r}\right) \sum_{i,l} f^l_i y^l_i$.*

Proof. Each facility i is opened with probability $r \cdot y_i$. The cost of installing services in step R5 is bounded by $\sum_i \Pr[i \text{ is opened (in R3 or R4)}] \sum_{l : y^l_i > 0} f^l_i = \sum_i r \cdot y_i \sum_{l : y^l_i > 0} f^l_i = r \cdot \sum_{i,l} f^l_i y^l_i$ since $y^l_i > 0 \implies y^l_i = y_i$.

Consider client $j \in D - D'$ with $g(j) = l$. For every non-central facility $i \in F_j$, let E_i be the event that i is opened in step R4 and $p_i = \Pr[E_i] = r \cdot y_i$. For every cluster center $k \in D'$ such that $S_k = F_j \cap N_k \ne \phi$, let E_k be the event that a facility from S_k is open after step R3. Let $p_k = \Pr[E_k] = \frac{\sum_{i \in S_k} y_i}{\sum_{i \in N_k} y_i} = r \cdot \sum_{i \in S_k} y_i$. Let m be the total number of events. All the events E_i are independent. The probability that service l is installed in step R6 due to client j, is the probability that no facility from F_j is open after steps R3 and R4, which is at most $\prod_{n=1}^{m} (1 - p_n) \le e^{-\sum_n p_n} = e^{-r}$. So the cost of installing services in step R6 is at most $\frac{1}{e^r} \sum_{j \in D - D'} f^{g(j)} \le \frac{1}{e^r} \sum_{i,l} f^l_i y^l_i$ since $\sum_{i \in F_j} y^l_i = 1$ and any two clients in D_l have disjoint F_j.

To bound the assignment cost, we bound the assignment cost incurred under a provably worse way of assigning demands to facilities. Demand j is assigned to a facility as follows. If some facility $i \in F_j$ is open, we assign demand j to the nearest such facility. Otherwise if $j \in D - D'$, j is assigned to its backup facility. If $j \notin D$, there is some client $k \in D_{g(j)} \subseteq D$ such that j was removed from $G_{g(j)}$ because a cluster was formed around k in step R1. We assign j to the same facility as k; so j may be assigned either to a facility in F_k or to its backup facility in $N_{\mathsf{nbr}(k)}$, if $k \notin D'$ and no facility from F_k is open. Note that service $g(j)$ is installed on the facility to which j is assigned. We need the following lemma from [6] (see also [15]).

Lemma 4.5. *Let $d_1 \le d_2 \le \ldots \le d_m$ and $0 \le p_n \le 1$ for $n = 1, \ldots, m$. Then,*

$$p_1 d_1 + (1-p_1) p_2 d_2 + \cdots + (1-p_1) \cdots (1-p_{m-1}) p_m d_m \le \frac{\sum_{n \le m} p_n d_n}{\sum_{n \le m} p_n} \Big(1 - \prod_{n \le m} (1 - p_n)\Big).$$

Lemma 4.6. *Let j be any demand. Let X be the distance between j and the facility assigned to it and Z be the event that no facility $i \in F_j$ is open. Then,* *(i) If $j \notin D'$, $\mathrm{E}\big[X|Z\big] \le 3\alpha_j + C_j(\gamma) + \bar{C}_j$, (ii) $\mathrm{E}\big[X\big] \le \bar{C}_j + \frac{1}{e^r}(3\alpha_j + C_j(\gamma))$.*

Proof. If $j \in D'$, $\mathrm{E}\big[X\big] = \sum_{i\in N_j} c_{ij}x_{ij} / \sum_{i\in N_j} x_{ij} \le \bar{C}_j$ since every facility in $F_j - N_j$ is farther from j than every facility in N_j. For $j \notin D'$, we show (i) and use it to prove (ii).

Suppose $j \in D - D'$, $k = \mathsf{nbr}(j)$ and $A = N_j \cap N_k \neq \phi$. For any facility $i \in A$ we have $c_{ij} \le \alpha_j$ and $c_{ik} \le C_k(\gamma)$. Let B be the distance between j and its backup facility in N_k. Event Z implies that j is assigned to the backup facility in N_k so conditioned on Z, $X = B$. If there is some $i \in A$ such that $c_{ik} \le \bar{C}_k$ we have a deterministic bound of $B \le \alpha_j + \bar{C}_k + C_k(\gamma)$. If there is no such i in A, since the unconditional distance between k and the backup facility in N_k is at most $\bar{C}_k$, by conditioning on Z we are only removing weight from facilities that are farther than the average distance. So the conditional expected distance between k and the backup facility is at most $\bar{C}_k$ implying that $\mathrm{E}\big[B|Z\big] = \mathrm{E}\big[X|Z\big] \le \alpha_j + C_k(\gamma) + \bar{C}_k$. In either case $\mathrm{E}\big[X|Z\big] \le \alpha_j + C_j(\gamma) + \bar{C}_j$, where the last inequality follows since we look at clients in D in increasing order of $C_j(\gamma) + \bar{C}_j$ and k was picked before j.

If $j \notin D$, there must be a client $k' \in D_{g(j)}$ such that j was removed from $G_{g(j)}$ because a cluster was formed around k' in step R1. So j, k' are assigned to the same facility, $2\alpha_{k'} + C_{k'}(\gamma) + \bar{C}_{k'} \le 2\alpha_j + C_j(\gamma) + \bar{C}_j$ and $c_{jk'} \le \alpha_j + \alpha_{k'}$. If a facility in $F_{k'}$ is open then we have a deterministic bound of $X \le \alpha_j + 2\alpha_{k'}$. Otherwise j, k' are assigned to the backup facility for k' and by the above bound on $\mathrm{E}\big[X|Z\big]$ for k', the conditional expected distance from j to the backup facility is at most $\alpha_j + 2\alpha_{k'} + C_{k'}(\gamma) + \bar{C}_{k'} \le 3\alpha_j + C_j(\gamma) + \bar{C}_j$.

We now prove part (ii). For a non-central facility $i \in F_j$, let p_i and E_i be as defined in Lemma 4.4 and let $d_i = c_{ij}$. For every $k \in D'$ such that $S_k = F_j \cap N_k \neq \phi$, let p_k, E_k be as defined in Lemma 4.4 and define $d_k = \mathrm{E}\big[\text{distance from } j \text{ to } S_k | E_k\big] = \sum_{i\in S_k} c_{ij}y_i / \sum_{i\in S_k} y_i$. Let the distances be ordered so that $d_1 \le d_2 \le \ldots \le d_m$ where there are m events in all. Since the events E_i are independent and $y_i = x_{ij}$, $p = \Pr[Z] = \prod_{n=1}^{m}(1-p_n) \le e^{-\sum_n p_n} = e^{-r}$. So,

$$\begin{aligned}
\mathrm{E}\big[X\big] \le\ & p_1 d_1 + (1-p_1)p_2 d_2 + \cdots \\
& + (1-p_1)\cdots(1-p_{m-1})p_m d_m + p \cdot \mathrm{E}\big[X|Z\big] \\
\le\ & \frac{\sum_{n\le m} p_n d_n}{\sum_{n\le m} p_n}(1-p) + p(3\alpha_j + C_j(\gamma) + \bar{C}_j) \\
\le\ & \bar{C}_j + \frac{1}{e^r}(3\alpha_j + C_j(\gamma)).
\end{aligned}$$

Theorem 4.2. *The randomized algorithm produces a solution of expected cost at most,*

$$\max\Big(r + \frac{4}{e^r}, 1 + \frac{1}{(1-\gamma)e^r} + \frac{3}{e^r}\Big) \cdot OPT$$

where $r = 1/\gamma$. Taking $\gamma = 0.67674$ we get a solution of cost at most $2.391 \cdot OPT$.

Proof. From the definition of $C_j(\gamma)$ and the Markov property we have, $C_j(\gamma) \le \frac{\bar{C}_j}{1-\gamma}$. The proof now follows from Lemmas 4.4 and 4.6.

5 The k-Median Variant

We consider a variant of this problem where we have the additional constraint that at most k facilities may be opened. This adds the constraint $\sum_i y_i \le k$ to the linear program (P). The objective function of the dual (D) gets modified to max $\sum_j \alpha_j - kz$ and constraint (2.2) changes to $\sum_j \beta_{ij} \le f_i + z$. Let (KP) and (KD) be the modified primal and dual programs and OPT_K be the value of an optimal k-median solution.

5.1 A modified primal-dual algorithm. We first modify the primal-dual algorithm of Section 3 to obtain the stronger guarantee that we return a solution to (P) of cost (O, I, C), and a solution (α, β, θ) to (D) such that $6O + I + C \le 6\sum_j \alpha_j \le 6 \cdot OPT$, where O, I, C denote respectively the facility opening cost, the service installation cost and the client assignment cost. We describe the algorithm briefly and sketch the analysis. The dual ascent process is the same as in Section 3 but we modify the way in which we open facilities and install services to ensure that a demand j does not pay for both opening a facility and for installing a service at some other facility. To do this we consider a more detailed notion of dependence between facilities. We classify 4 types of dependence between facilities. Say that the *ordered pair* (i, i') is,

(1) ff-dependent (f for facility) if there is a demand j such that $\beta_{ij}, \beta_{i'j} > 0$.

(2) sf-l dependent (s for service) if there exists $j \in G_l$ such that $\theta_{ij}, \beta_{i'j} > 0$.

(3) ss-l dependent if for some $j \in G_l$, both $\theta_{ij}, \theta_{i'j} > 0$.

(4) fs-l dependent if for some $j \in G_l$, both $\beta_{ij}, \theta_{i'j} > 0$.

Recall that F is the set of tentatively open facilities, $F_l \subseteq F$ the set of tentatively open facilities on which service l is tentatively installed, t_i is the time at which facility $i \in F$ became tentatively open, and for $i \in F_l$, t_{il} is the time at which service l was tentatively installed at i. Initially for each facility $i \in F$, let S_i be the set of services that are tentatively installed at i.

M1. We first pick a set $F' \subseteq F$ of facilities to open, and for each $i \in F'$ a set $T_i \subseteq S_i$ of services to install at facility i. Initially $F' = \phi$ and $T_i = \phi$ for all i. We consider facilities in F in the order given by $\mathcal{O}$. While $F \neq \phi$,

1. Let $i \in F$ be the currently considered facility. Remove i from F, set $F' \leftarrow F' \cup \{i\}$, $T_i = S_i$.
2. For each $i' \in F$ we do the following.
   a) If (i, i') is ff-dependent OR $\exists l \in T_i$ s.t. (i, i') is sf-l dependent OR $\exists l \in S_{i'}$ s.t. (i, i') is fs-l dependent and $t_{i'l} < t_{i'}$, set $F \leftarrow F - \{i'\}$. Call i the neighbor of i' and denote it by $\mathsf{nbr}(i')$. Otherwise,
   b) For every $l \in S_{i'}$, if (i, i') is fs-l dependent (so $t_{i'l} \geq t_{i'}$) OR $l \in T_i$ and (i, i') is ss-l dependent, set $S_{i'} \leftarrow S_{i'} - \{l\}$.

We open the facilities in F' and for each $i \in F'$ install all of the services in T_i at i.

M2. We now install services at some more facilities. Consider service type l. Let A_l be the facilities in F' at which service l is installed (i.e., $l \in T_i$). Note that $A_l \subseteq F' \cap F_l$. Let $B_l = F_l - F'$. We remove from B_l every facility i' for which there is some facility $i \in F'$ such that (1) (i, i') is fs-l dependent, OR (2) $i \in A_l$ and (i, i') is ss-l dependent. We say that a set of facilities is ss-l independent if no pair (i, i') of facilities from the set is ss-l dependent. We pick a maximal ss-l independent subset $F'_l \subseteq B_l$. Initially $F'_l = \phi$. We consider facilities in B_l in increasing order of t_i and add facility i to F'_l if $F'_l \cup \{i\}$ remains ss-l independent. For every $i \in F'_l$ we install service l on facility $\mathsf{nbr}(i) \in F'$.

M3. Each client j is assigned to the nearest open facility at which service $g(j)$ is installed.

Analysis Sketch. Consider the set of demands $D' = \{j : \exists i \in F' \text{ s.t. } \beta_{ij} > 0\}$. By the construction of F', we know that for each demand j there is at most one $i \in F'$ such that $\beta_{ij} > 0$; for $j \in D'$ let $o(j)$ denote this unique facility in F'. By design we ensure that any demand $j \in G_l$ contributes $\theta_{ij} > 0$ for at most one facility $i \in A_l \cup F'_l$ and further that if $j \in D'$ then $i = o(j)$ is the only such facility. This gives the following.

Lemma 5.1. *The cost of opening facilities is $\sum_{j \in D'} \beta_{o(j)j}$. The cost of installing services is at most $\sum_{j \in D'} \theta_{o(j)j} + \sum_{j \notin D'} \alpha_j$.*

The bound on the assignment cost incurred is similar to the bound in Lemma 4.3 and is proved similarly.

Lemma 5.2. *If $j \in D'$, the assignment cost of j is at most $3(\alpha_j - \beta_{o(j)j})$. If $j \notin D'$ the assignment cost incurred for j is at most $5\alpha_j$.*

Combining the above two lemmas and the fact that for $j \in D'$, $\alpha_j = c_{o(j)j} + \beta_{o(j)j} + \theta_{o(j)j}$, we get the following theorem.

Theorem 5.1. *The solution returned has cost (O, I, C) such that $6O + I + C \leq 6 \sum_j \alpha_j \leq 6 \cdot OPT$.*

5.2 An 18-approximation algorithm. Suppose we fix z and run the above primal-dual algorithm with the facility costs modified to $f_i + z$. Suppose the algorithm returns a primal solution of cost (O, I, C) that opens k facilities and a dual solution (α, β, θ). Here O is the facility opening cost with the original costs f_i. Then, we can show that we have a solution of cost at most $6 \cdot OPT_K$. since $(\alpha, \beta, \theta, z)$ is a feasible solution to (KD) and by Theorem 5.1, $6(O+kz)+I+C+ \leq 6\sum_j \alpha_j \implies 6O+I+C \leq 6(\sum_j \alpha_j - kz) \leq 6 \cdot OPT_K$. So our goal is to find such a z in polynomial time. We do not quite know how to do this, instead we find two values $z_1 > z_2$, close together such that the algorithm opens $k_1 < k$ facilities for z_1 and $k_2 > k$ facilities for z_2. When z is very large, e.g., $|\mathcal{D}|(\max_{ij} c_{ij} + \max_{il} f_i^l)$, the algorithm will open just one facility and at $z = 0$ the algorithm opens $> k$ (we assume this — otherwise the solution at $z = 0$ costs at most $6 \cdot OPT_K$ since $(\alpha, \beta, \theta, 0)$ is a feasible dual solution). We perform a bisection search in this range to find z_1 and z_2. We combine these two solutions to first get a fractional solution of cost $6(1-\epsilon)^{-1} \cdot OPT_K$, for some small ϵ, that (fractionally) opens k facilities and in which each demand is assigned to at most two facilities. If the service cost depends only on the service type, we can round this solution losing a factor of $3(1-\epsilon)$ to get an integer solution that opens exactly k facilities, thus getting an overall approximation ratio of 18. This idea was used in [9] for the k-median variant of UFL. However our final rounding procedure differs from the one in [9] since we also have to pay for installing services and need to ensure that demand j is connected to an open facility at which service $g(j)$ is installed.

Theorem 5.2. *If the service installation cost depends only on the service and not on the location, we can get an 18-approximation algorithm for the k-median variant of facility location with service installation costs.*

6 Extensions

Arbitrary Demands. Our results carry over to the case where instead of unit demands, client j may have a demand $d_j \geq 0$. We can reduce this to the unit demand case by making d_j copies of client j, but this makes the algorithm run in pseudo-polynomial time. We can simulate this reduction however. In the primal-dual algorithm we raise α_j at rate d_j and say that j has reached i if $\alpha_j \geq d_j c_{ij}$. In the LP rounding algorithms of Section 4 the only change is that α_j gets replaced with α_j/d_j in steps A1, A2 and R1. The analysis in Sections 3, 4 and 5 extends in a straightforward way and we get the same approximation guarantees.

Acknowledgments. We thank Robin Roundy for stimulating discussions and helpful suggestions.

References

[1] N. Aksoy and S. Erenguc. Multi-item inventory models with coordinated replenishments: a survey. *International Journal of Operations & Production Management*, 8:63–73, 1988.

[2] A. Archer, R. Rajagopalan and D. Shmoys. Lagrangian relaxation for the k-median problem: new insights and continuity properties. In *Proceedings of 11th ESA*, pages 31–42, 2003

[3] I. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 661–670, 2001.

[4] Y. Bartal, M. Charikar and D. Raz. Approximating min-sum k-clustering in metric spaces. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 11–20, 2001.

[5] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In *Proceedings of 5th IPCO*, pages 180–194, 1998.

[6] F. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*. To appear.

[7] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

[8] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. Greedy facility location algorithms analyzed using dual-fitting with factor-revealing LP. *Journal of the ACM*. To appear.

[9] K. Jain and V.V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48:274–296, 2001.

[10] J. H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 771–782, 1992.

[11] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location. In *Proceedings of 5th APPROX*, pages 229–242, 2002.

[12] P. Mirchandani and R. Francis, eds. *Discrete Location Theory*. John Wiley and Sons, Inc., New York, 1990.

[13] D. B. Shmoys. Approximation algorithms for facility location problems. In *Proceedings of 3rd APPROX*, pages 27–33, 2000.

[14] D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 265–274, 1997.

[15] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *Proceedings of 9th IPCO*, pages 240–257, 2002.

[16] C. Swamy. A note on the data placement problem. Unpublished manuscript, 2003.