Greedy strikes back: Improved Facility Location Algorithms

Sudipto Guha*
Dept. of Computer Science
Stanford University
Stanford, CA 94305

Samir Khuller†
Computer Science Dept. and UMIACS
University of Maryland
College Park, MD 20742

Abstract

A fundamental facility location problem is to choose the location of facilities, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the facilities, as well as transportation costs for distributing the commodities. This problem is commonly referred to as the *uncapacitated facility location* problem. Applications to bank account location and clustering, as well as many related pieces of work, are discussed by Cornuejols, Nemhauser and Wolsey [2]. Recently, the first constant factor approximation algorithm for this problem was obtained by Shmoys, Tardos and Aardal [14].

We show that a simple greedy heuristic combined with the Shmoys, Tardos and Aardal algorithm [14], can be used to derive a better approximation guarantee. We discuss a few variants of the problem, demonstrating better approximation factors for restricted versions of the problem.

We also show that the problem is Max SNP-hard. However, the inapproximability constants derived from the Max SNP hardness are very close to one. By relating this problem to Set Cover, we can prove a better lower bound on the best possible approximation ratio.

1 Introduction

We consider the following facility location problem: given a set of locations F such that we may build a facility at location i with cost f_i, and a set of demand locations D, where clients require service. If a client at location j with demand d_j is assigned to a facility at location i we incur a service cost of $d_j \cdot c_{ij}$. Our objective is to minimize the *total* cost of building facilities and servicing the clients. We assume that the service costs c_{ij} are symmetric and form a metric. Not surprisingly, this problem is NP-hard.

An approximation algorithm with a factor of ρ is a polynomial time algorithm that produces a solution whose total cost is at most ρ times the optimal solution cost. Hochbaum [4] showed that this can be approximated with a factor of $O(\log n)$ by using a greedy algorithm. Very recently, Shmoys, Tardos and Aardal [14] gave the first constant factor approximation algorithm that achieves an approximation ratio of 3.16. This algorithm is based on a method introduced by Lin and Vitter [8] for rounding fractional solutions to linear programming relaxations. Recently and independently, Korupolu, Plaxton and Rajaraman [6] have shown that the local search heuristic proposed by Kuehn and Hamburger [7] achieves a constant approximation for many variants of the UFL problem.

Uncapacitated facility location (UFL) problems have been *widely studied* in the *Operations Research* literature (see [1, 7, 10, 15]). This problem has been studied from different perspectives, as clustering analysis [12], machine scheduling and information retrieval, communication networks [11] and many others. For an extensive survey of closely related work one can read the chapter by Cornuejols, Nemhauser and Wolsey [2].

Outline of Results

It is easy to show that the problem is Max SNP-hard (Subsection 3.1). We prove that if there is a polynomial time algorithm that approximates the problem to a factor better that 1.463 then $NP \subseteq DTIME[n^{O(\log\log n)}]$. This hardness result is established through hardness proofs for the set cover problem [9, 3] (Subsection 3.2).

We also show that a simple and natural greedy "local-improvement" algorithm combined with the algorithm given by Shmoys, Tardos and Aardal [14] yields an approximation factor of 2.408. This algorithm and its proof are described in Section 4. In Section 5 we present a simple family of examples for which the "gap" between the integral and fractional solution for the LP formulation given by Shmoys, Tardos and Aardal [14] is a factor of 1.463.

*Supported by an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email addr: `sudipto@cs.stanford.edu`

†Research supported by NSF CAREER Award CCR-9501355. Email addr: `samir@cs.umd.edu`

The hardness results are proven assuming that the facility costs for locations in F are identical. In this special case when facility costs are uniform, we can improve the approximation ratio to 2.225 (Subsection 4.4).

In Section 6 we consider a model where every location is a facility and demand vertex. We refer to this problem as the *complete metric uncapacitated facility location problem.* We use the same techniques to show that this simplified version when facilities can be built anywhere, and all facilities cost the same – has an inapproximability bound of 1.278 unless $NP \subseteq DTIME[n^{O(\log \log n)}]$. For this model, where facilities can be built at any location at unit cost, we give an algorithm which achieves an approximation guarantee of 2.104. Thus we demonstrate varying degrees of hardness and possible approximation factors for various variations of the basic problem.

2 Preliminaries

In this section we discuss the basic notation used in the Shmoys, Tardos, Aardal [14] paper. We are given a set of locations $N = \{1, \dots, n\}$ and a subset $F \subseteq N$ of locations at which we may open a facility, and a subset $D \subseteq N$ of locations that must be assigned to an open facility. For each location $j \in D$ there is a positive demand d_j that must be shipped from its assigned location. For each location $i \in F$ the cost of opening a facility is f_i. The service cost of assigning location $j \in D$ to an open facility at i is c_{ij} per unit of demand shipped. We assume that the c_{ij} costs are non-negative, symmetric and satisfy the triangle inequality. We wish to find an assignment of each location in D to an open facility so as to minimize the total cost incurred. This is the *uncapacitated facility location problem* (UFL).

We can state the above problem as an integer programming problem as in [14, 1]. There is a 0/1 variable $y_i, i \in F$ indicating if a facility is open at location i. The 0/1 variable $x_{ij}, i \in F, j \in D$ indicates if client j is assigned to the facility at i.

$$\min \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in D} d_j c_{ij} x_{ij}$$

subject to
$$\sum_{i \in F} x_{ij} = 1 \text{ for each } j \in D$$
$$x_{ij} \le y_i \text{ for each } i \in F, j \in D$$
$$x_{ij} \in \{0,1\} \text{ for each } i \in F, j \in D$$
$$y_i \in \{0,1\} \text{ for each } i \in F$$

Shmoys, Tardos and Aardal [14] derive an algorithm that is based on solving a linear relaxation of the above integer program, and rounding the fractional solution to an integer solution.

Let us define λ_f to be the cost of opening facilities in the optimal solution, and λ_c to be the service cost corresponding to this optimal solution.

Our linear programming formulation is the same, with the extra constraint that bounds the fractional cost of opening facilities. In other words, we add the new constraint that
$$\sum_{i \in F} f_i y_i \le \lambda_f$$
for a given value λ_f which is the "budget" for the cost of constructing centers. Once we do this, we can eliminate the facility construction cost from the objective function, and simply minimize the service cost. The reason for doing this will become clear when we discuss the algorithm. Notice that after adding this constraint, we still have a fractional solution whose service cost is at most λ_c and facility construction cost is at most λ_f.

3 Hardness Results

3.1 Max SNP hardness The following problem was shown to be Max SNP-hard by Papadimitriou and Yannakakis [13].

B-Vertex Cover: Given a graph $G = (V, E)$, with the degree of each node bounded by a constant B, find the minimum cardinality vertex cover.

THEOREM 3.1. *The Uncapacitated Facility Location problem is Max SNP-hard.*

We prove this result by reducing from B-vertex cover.

Proof: The B-vertex cover problem has has been shown to be Max SNP-hard by Papadimitriou and Yannakakis [13]. We will show that a $(1+\epsilon')$ approximation algorithm for UFL implies a $(1+\epsilon)$ approximation for B-vertex cover. Construct an instance of the UFL problem where the vertices of G correspond to facility locations, and the edges correspond to demand locations. The distance (service cost) between facility i (representing vertex i) and demand location j (representing edge j) is 1 if edge j is incident on vertex i, and 2 otherwise. Let the optimal vertex cover have size k. The cost of opening a facility is $\frac{|E|}{Bk}$. All demands are one.

Suppose that the UFL problem can be approximated within a factor of $1+\epsilon'$, ($\epsilon' = \frac{\epsilon}{1+B}$). Corresponding to the vertex cover optimal solution, there exists an assignment of cost at most $(|E| + \frac{|E|}{B})$. We are guaranteed a solution of cost at most $(|E| + \frac{|E|}{B})(1 + \frac{\epsilon}{1+B}) = |E| + (1+\epsilon)\frac{|E|}{B}$.

Suppose the algorithm chooses βk facilities and covers $t|E|$ edges at distance exactly 2. The total cost is $\beta \frac{|E|}{B} + |E| + t|E|$. This cost can be bounded by

$|E| + (1+\epsilon)\frac{|E|}{B}$, thus $\beta + tB \leq (1+\epsilon)$.

By choosing the vertices corresponding to the open facilities and one vertex for each of $t|E|$ edges, we obtain a vertex cover. Since $Bk \geq |E|$, the size of the vertex cover is $\beta k + t|E| \leq (\beta + tB)k \leq (1+\epsilon)k$. □

Since the best known lower bound for the vertex cover problem is $\frac{7}{6}$, this does not give very strong hardness bounds. We present a scheme to produce much stronger lower bounds, based on the recent results on the hardness of approximating Set Cover [3].

3.2 Improved Hardness Results We establish a relationship between the Set Cover problem and the UFL problem, and show that if there is a polynomial time algorithm for the UFL problem with an approximation factor smaller than 1.463 then we have a polynomial time approximation algorithm for Set Cover that has an approximation ratio of $c \ln |X|$ for some $c < 1$. By Feige's [3] recent result, this implies that $NP \subseteq DTIME[n^{O(\log \log n)}]$.

Consider an instance of Set Cover defined as follows: let the set of elements be $X = \{x_1, \ldots, x_n\}$. Let the collection of sets be defined as $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where each $S_i \subseteq X$. Moreover, we may assume that k is the size of the minimum set cover. (Since $1 \leq k \leq m$ we can always run any set cover algorithm for each possible value of k.)

LEMMA 3.1. *Suppose we have a set cover instance $(X, \mathcal{S})$ with the minimum set cover size being k. If there is a polynomial time algorithm that can pick βk sets (for any constant β) and cover $c' \cdot |X|$ elements, where $c' > c_\beta = (1 - \frac{1}{e^\beta})$ then $NP \subseteq DTIME[n^{O(\log \log n)}]$, where $n = |X|$.*

Proof: We run the algorithm to find a partial covering. We delete the covered elements and sets that were picked, and run the algorithm again on the remaining elements. Suppose in iteration i, the number of uncovered elements at the start of the iteration is n_i. The algorithm picks $\beta_i k$ sets and covers $c_i n_i$ elements, where $c_i > (1 - \frac{1}{e^{\beta_i}})$. Clearly, $n_{i+1} = n_i(1 - c_i)$ and $n_1 = |X|$. Suppose after ℓ iterations $n_{\ell+1} = 1$. The total number of sets that are picked is $\sum_{i=1}^{\ell} \beta_i k$, with an approximation factor of $\sum_{i=1}^{\ell} \beta_i$.

$$n_{\ell+1} = 1 = |X| \prod_{i=1}^{\ell} (1 - c_i).$$

$$\ln |X| = \sum_{i=1}^{\ell} \ln \frac{1}{1 - c_i}.$$

Using the fact that $c_i > (1 - \frac{1}{e^{\beta_i}})$, we get for each i,

$$\beta_i < \ln \frac{1}{1 - c_i}.$$

$$\beta_i \leq c \cdot \ln \frac{1}{1 - c_i} \text{ for some } c < 1.$$

$$\sum_{i=1}^{\ell} \beta_i \leq c \sum_{i=1}^{\ell} \ln \frac{1}{1 - c_i} = c \ln |X| \text{ for some } c < 1.$$

By Feige's result [3] this implies that $NP \subseteq DTIME[n^{O(\log \log n)}]$. □

The above lemma essentially argues that by picking β times the number of sets in an optimal cover, we cannot cover more than c_β fraction of the elements.

THEOREM 3.2. *If there is a polynomial time algorithm with an approximation factor smaller than* 1.463 *for the* metric uncapacitated facility location problem *then $NP \subseteq DTIME[n^{O(\log \log n)}]$.*

Proof:

We transform an instance of set cover $(X, \mathcal{S})$ to an instance of the UFL problem. We set $D = X$ and $F = \mathcal{S}$. Each f_i value (facility construction cost) is identical. If $x_j \in S_i$ then $c_{ij} = 1$. (In other words, distances are 1 to the sets (facilities) that this element (client) belongs to.) We extend c_{ij} to be a metric using shortest paths. We set $d_j = 1$ for each client j.

The basic idea is to use the approximation algorithm for solving UFL to find partial solutions for the set cover instance, by repeatedly running the algorithm. If the algorithm repeatedly finds a small partial cover, covering a large fraction of the elements, then by Lemma 3.1 we can show that we can argue that $NP \subseteq DTIME[n^{O(\log \log n)}]$. Otherwise, if ever the algorithm does not cover a large fraction of the elements, we obtain a lower bound on the approximation ratio.

We now describe the proof more formally. Consider the following algorithm to solve an instance of Set Cover. Let γ be a constant whose value will be specified later.

SET COVER $(X, \mathcal{S})$
Create a UFL instance (F, D) corresponding to $(X, \mathcal{S})$
Let $f = \gamma \frac{|D|}{k}$ (cost of a facility)
while $D \neq \emptyset$ **do**
    F' = UFL-Approx-Algorithm (F, D)
    Let D' be the clients covered with cost 1
    Let $F = F - F'$ and $D = D - D'$
    Let $f = \gamma \frac{|D|}{k}$ be the cost of a facility
end-proc

Suppose that in iteration j, $n_j = |D|$ and f_j is the cost of a facility. Since there is a solution of size k to the set cover problem that covers all the elements, there is a solution to the facility location problem of cost $f_j k + n_j$ (we build facilities at the k locations corresponding to the set cover, all the service costs are 1, recall $|D| = n_j$).

If there is an α approximation then we are guaranteed to obtain a solution of cost at most $\alpha(f_j k + n_j)$.

We will choose the value of f_j so that $f_j = \gamma\frac{n_j}{k}$. Our total cost is at most $\alpha n_j(1+\gamma)$.

We now consider a solution obtained by the approximation algorithm. Suppose we open $\beta_j k$ centers at cost f_j each. Suppose the number of clients that are within distance 1 of these centers is $c_j n_j$. All other clients (uncovered elements) are at distance at least 3. The cost of this solution is at least

$$f_j\beta_j k + c_j n_j + 3(n_j - c_j n_j).$$

This cost is at most $\alpha n_j(1+\gamma)$. Thus we obtain (after dividing by n_j)

$$\alpha(1+\gamma) \geq \gamma\beta_j + 3 - 2c_j.$$

$$\alpha \geq \frac{\gamma\beta_j + 3 - 2c_j}{(1+\gamma)}.$$

If for any j, $c_j \leq c_{\beta_j}$ then

$$\alpha \geq \frac{\gamma\beta_j + 3 - 2c_{\beta_j}}{(1+\gamma)}.$$

If we fix γ, and compute the minimum value as a function of β_j we can compute the smallest value for the RHS. This will certainly be a lower bound for α, our approximation ratio. Taking the derivative with respect to β_j, tells us that the minimum is achieved at $\beta_j = \ln\frac{2}{\gamma}$. Substituting this value of β_j gives us

$$\alpha \geq 1 + \frac{\gamma}{1+\gamma}\ln\frac{2}{\gamma}.$$

Choosing $\gamma = 0.463$ gives $\alpha > 1.463$.

If for each j, $c_j > c_{\beta_j}$ then we have an algorithm that picks $\beta_j k$ sets and covers $c_j n_j$ elements each time we run the algorithm. By Lemma 3.1 we conclude that SET COVER $(X, \mathcal{S})$ is an algorithm that gives an approximation guarantee of $c \ln n$ for $c < 1$, thus $NP \subseteq DTIME[n^{O(\log\log n)}]$.

□

Remark: Notice the above proof uses a distance metric where the distances are either 1 or 3. For such a distance metric of 1 or 3 and arbitrary costs we can develop an algorithm that achieves an approximation ratio of 1.463 (Corollary 4.1). For uniform costs and arbitrary distances we present an algorithm with an approximation factor of 2.225 (Subsection 4.4). It would be very interesting if we can use non-uniform costs and distances to get an improved lower bound for the general problem.

4 Improved Approximation Algorithm

4.1 Background Review We consider the linear programming relaxation of the integer program as discussed in Section 2. We can guess the facility cost λ_f of the optimum solution, within a factor of $(1+\epsilon)$ for any fixed ϵ.

We add the following constraint

$$\sum_{i\in F} f_i y_i \leq \lambda_f.$$

The rest of this section will assume that we have guessed the facility cost approximately, to within a $1+\epsilon$ factor. Once we have a budget on the facility construction cost, we can simplify the objective function to only include the service cost. Consider the fractional solution corresponding to this new LP. The service cost of the fractional solution is at most the service cost of the optimal of the integer program. Let facility i serve a demand location j, iff $x_{ij} > 0$. For each demand location j, order the facilities that serve location j in increasing order of service cost c_{ij}.

We now outline the algorithm given by Shmoys, Tardos and Aardal [14]. For every α, define $N_\alpha(j)$ as the set of facilities with the lowest service costs for j and serving j such that their y_i values sum to at least α. That is $\sum_{s=1}^{i'} y_s \geq \alpha$ and $\sum_{s=1}^{i'-1} y_s < \alpha$. The threshold of j, denoted by $\tau(\alpha, j)$, is defined as the service distance to the farthest facility in the set $N_\alpha(j)$. Recursively choose the unsatisfied demand node with the minimum $\tau(\alpha, j)$ value. Consider the cheapest facility in $N_\alpha(j)$, for this demand node. By opening a facility there, the cost goes up by at most a factor of $\frac{1}{\alpha}$. If for any node j', $N_\alpha(j) \cap N_\alpha(j')$ is nonempty, serve j' by the facility chosen by node j. Mark j and all such j' as satisfied.

It is easy to argue that every node j' gets served within service distance $3\tau(\alpha, j')$. A node which was chosen with minimum $\tau(\alpha, j)$ value, it is trivially true. A node j' that was marked by j can be served within service distance $\tau(\alpha, j') + 2\tau(\alpha, j) \leq 3\tau(\alpha, j')$. This follows from the minimality of $\tau(\alpha, j)$.

Since each open facility corresponds to disjoint $N_\alpha(j)$, the cost of facilities is at most $\frac{1}{\alpha}\lambda_f$. The service cost is at most $\sum_j 3d_j\tau(\alpha, j)$.

Define

$$g(\alpha) = \frac{\sum_j d_j\tau(\alpha, j)}{\lambda_c}$$

Thus the algorithm developed by the Shmoys, Tardos, Aardal [14] guarantees a facility construction cost of at most $\frac{1}{\alpha}\lambda_f$ and a service cost of at most $3g(\alpha)\lambda_c$.

LEMMA 4.1. *Function g satisfies the property that $\int_0^1 g(\alpha)d\alpha \leq 1$.*

Proof:

$$\int_0^1 g(\alpha)d\alpha = \sum_j \frac{d_j}{\lambda_c}\int_0^1 \tau(\alpha, j)d\alpha = \sum_j\left(\frac{d_j}{\lambda_c}\sum_i x_{ij}\cdot c_{ij}\right)$$
$$= \frac{1}{\lambda_c}\sum_j \text{ cost for } j \text{ in fractional solution } \leq 1$$

This establishes the claim. $\square$

Observe that g is a monotonically increasing step function with at most polynomially many steps. (This will be used later to design a better algorithm.)

4.2 Return of the Greedy By an (a, b) approximation we denote a solution with facility cost at most $a\lambda_f$, and service cost at most $b\lambda_c$.

In [14] a $\left(\frac{1}{\alpha}, 3g(\alpha)\right)$ approximation is used to give an algorithm with cost 3.16 times the optimal.

Assume we have a bicriteria (a, b) approximation for the UFL problem. Assume that $b \geq 1 + \frac{\lambda_f}{\lambda_c}$. We provide a scheme that will give us a solution with a different approximation guarantee. *The key idea is to open more facilities to reduce the service cost.*

Let $S(F)$ denote the minimum service cost associated with set F of facilities.

$$S(F) = \sum_{j \in D} d_j \cdot \text{shortest path from } j \text{ to a member of } F$$

Greedy Improvement Algorithm(α)
$F =$ Initial Set of Facilities formed by STA(α)
Repeat
Step 1. Consider an unused facility v
Let $Gain(v) = S(F) - S(F \cup \{v\}) - f(v)$.
(Recall $f(v)$ is the cost of the facility v.)
Step 2. Find the facility v' with the highest (positive) $\frac{Gain(v')}{f(v')}$ value.
Step 3. $F = F \bigcup \{v'\}$
Until no facility has positive gain.
Return F as the set of facilities.

THEOREM 4.1. *The total cost paid by the algorithm is at most* $\lambda_f \left(a + \ln \frac{b-1}{\frac{\lambda_f}{\lambda_c}} + 1\right) + \lambda_c$.

Proof: At the start of iteration i let the set of facilities be F_i.

Suppose for an iteration i the service cost $S(F_i)$ exceeds $\lambda_c + \lambda_f$. If we add the facilities used by the optimal solution, the service cost reduces to λ_c. Thus there exists a set of facilities whose total cost is λ_f and total gain of these facilities is $S(F_i) - \lambda_c - \lambda_f$.

LEMMA 4.2. *At the start of iteration i, there exists a facility whose gain to cost ratio is at least* $\frac{S(F_i) - \lambda_c - \lambda_f}{\lambda_f}$.

Proof: Suppose each facility has a gain to cost ratio that is strictly smaller than $\Gamma = \frac{S(F_i) - \lambda_c - \lambda_f}{\lambda_f}$. Let OPT_F be an optimal set of facilities, with total facility cost λ_f and service cost λ_c.

If we add a facility j in OPT_F to the current set of facilities F_i, let the gain be g_j. Notice that if we add facility j to a set of facilities that contain F_i then the gain is only smaller. Clearly, $g_j < \Gamma \cdot f_j$ where f_j is the cost of facility j. Suppose we add the facilities in OPT_F one at a time (in increasing facility number). Let $\Delta S(F_i^j)$ be the change of the service cost when facility j is added to $F_i \cup \{1, \ldots, j-1\}$. By definition of g_j, $\Delta S(F_i^j) - f_j \leq g_j < \Gamma \cdot f_j$. Thus,

$$\Delta S(F_i^j) < (1 + \Gamma) f_j.$$

By summing (over all j) and replacing the value of Γ we obtain

$$\sum_{j \in OPT_F} \Delta S(F_i^j) < (S(F_i) - \lambda_c) \frac{\sum_{j \in OPT_F} f_j}{\lambda_f} \leq S(F_i) - \lambda_c.$$

This is a contradiction, since the service cost drops by at least $S(F_i) - \lambda_c$ when we add all the facilities in OPT_F. $\square$

Let us denote the cost of the set of facilities F_i by $C(F_i)$. Since we chose the facility with the maximum gain to cost ratio,

$$\frac{S(F_i) - S(F_{i+1})}{C(F_{i+1}) - C(F_i)} = \frac{Gain(v') + f(v')}{f(v')}$$
$$\geq \frac{S(F_i) - \lambda_c - \lambda_f}{\lambda_f} + 1 = \frac{S(F_i) - \lambda_c}{\lambda_f}.$$

This tells us that we stop only after our service cost has decreased below $\lambda_c + \lambda_f$. Suppose we cross this threshold at iteration ℓ. Using the fact that the total cost does not increase, it is sufficient to show that the theorem holds at the end of the iteration ℓ. Rearrange the last equation as,

(4.1)
$$S(F_{i+1}) \quad \leq \quad S(F_i) - (S(F_i) - \lambda_c) \left(\frac{C(F_{i+1}) - C(F_i)}{\lambda_f} \right)$$

Subtracting λ_c from both sides, and observing that $1 - x \leq e^{-x}$ for $x \leq 1$,

$$\begin{aligned} S(F_{i+1}) - \lambda_c &\leq (S(F_i) - \lambda_c)\left(1 - \frac{C(F_{i+1}) - C(F_i)}{\lambda_f}\right) \\ &\leq (S(F_i) - \lambda_c) \cdot e^{-\frac{C(F_{i+1}) - C(F_i)}{\lambda_f}} \end{aligned}$$

The solution to this recurrence, for any j, is

$$S(F_j) \leq \lambda_c + (S(F_1) - \lambda_c) e^{-\frac{C(F_j) - C(F_1)}{\lambda_f}}$$

Notice that $C(F_1) \leq a\lambda_f$, and $S(F_1) \leq b\lambda_c$. Hence the algorithm certainly halts after $C(F)$ exceeds $C(F_1) + \lambda_f \cdot \ln \frac{(b-1)\lambda_c}{\lambda_f}$. For this facility construction cost, the service cost is guaranteed to decrease below $\lambda_c + \lambda_f$.

However this does not give the approximation bound desired, because the $C(F)$, *could have exceeded*

$C(F_1) + \lambda_f \cdot \ln \frac{(b-1)\lambda_c}{\lambda_f}$. This situation occurs at the last iteration of the algorithm, and that if we exceed this, then the distance cost also does go down accordingly.

Let us assume the algorithm stopped after iteration ℓ. Therefore $S(F_\ell) \geq \lambda_c + \lambda_f$, rewrite this for some $x > 1$,

$$S(F_\ell) = \lambda_c + x\lambda_f$$

Let the cost paid in the last iteration be $\lambda_f \Delta_c$. (Note that $\Delta_c \leq 1$, since we can ignore facilities whose cost exceeds λ_f.) From equation(4.1) we conclude that $S(F_{\ell+1}) \leq S(F_\ell) - \Delta_c(S(F_\ell) - \lambda_c)$. Rewrite this as for some $\mu \geq 1$,

$$S(F_{\ell+1}) = S(F_\ell) - \mu\Delta_c(S(F_\ell) - \lambda_c) = \lambda_c + \lambda_f x(1 - \mu\Delta_c)$$

Since $S(F_{\ell+1}) \leq \lambda_c + \lambda_f$, we get the following relations between x, Δ_c and μ,

$$x(1 - \mu\Delta_c) \leq 1 \iff -\frac{1}{x} + 1 - \mu\Delta_c \leq 0$$

The cost paid at the earlier iterations can be bounded as

$$\begin{aligned} C(F_\ell) &\leq C(F_1) + \lambda_f \cdot \ln \frac{S(F_1) - \lambda_c}{S(F_\ell) - \lambda_c} \\ &\leq \left(a + \ln \frac{b-1}{x\frac{\lambda_f}{\lambda_c}} \right) \lambda_f. \end{aligned}$$

Thus the total cost paid by us is at most $C(F_\ell) + S(F_{\ell+1}) + \lambda_f \Delta_c$, which is at most

$$\left(a + \ln \frac{b-1}{\frac{\lambda_f}{\lambda_c}} \right) \lambda_f + \lambda_c + \lambda_f(-\ln x + x - \mu x \Delta_c + \Delta_c).$$

The derivative w.r.t x of the third term in the summand is, $-\frac{1}{x} + 1 - \mu\Delta_c$. Notice that this is negative due to relations between x, Δ_c and μ. The term therefore is greatest when x approaches 1, but is $1 - (\mu - 1)\Delta_c$, and hence at most 1. This proves the theorem.

□

Corollary 4.1. *For a metric with distances* 1 *and* 3 *we have an approximation with* $a = 0$ *and* $b = 3$, *since picking any center bounds the service cost by 3. From the above formula it is a* $1 + \frac{\gamma}{\gamma+1} \ln \frac{2}{\gamma}$ *approximation if we choose facilities greedily. This is a* 1.463 *approximation.*

4.3 Improved Approximation Algorithm We combine the ideas in the above section to present the full algorithm.

Final Algorithm(λ_f)

Step 1. Solve the LP formulated in Section 2.

Notice that $g(\alpha)$ is a step function with polynomially many steps.

Step 2. For each α where the function $g(\alpha)$ has a step, run *Greedy Improvement Algorithm*(α).

Step 3. Choose the solution which minimizes the cost in Step 2.

Theorem 4.2. *The above algorithm yields a solution of cost at most* 2.408 *times the optimal.*

Proof: Let $c = 2.408$ and assume the contrary. Let $\gamma = \frac{\lambda_f}{\lambda_c}$. For an $(\frac{1}{\alpha}, 3g(\alpha))$ approximation, the facility cost is at most $\frac{\lambda_f}{\alpha} = \frac{\lambda_c \gamma}{\alpha}$ and the service cost is at most $3g(\alpha)\lambda_c$.

If for any value of α tried by the algorithm, we have $3g(\alpha)\lambda_c \leq \lambda_c(c(\gamma+1) - \frac{\gamma}{\alpha})$, we immediately obtain a factor c approximation. Thus w.l.o.g.,

$$3g(\alpha) > c(1 + \gamma) - \frac{\gamma}{\alpha}.$$

Since $g(\alpha)$ is a monotonic step function, the inequality is valid for every α where a step occurs, we can extend the inequality to hold for all α. This inequality is only meaningful for $\alpha \geq \frac{\gamma}{c(1+\gamma)}$.

When $\alpha \geq \frac{\gamma}{(c-1)(1+\gamma)}$, we have, $3g(\alpha) > (1 + \gamma)$. For this value of b, we can use the greedy algorithm to get cost at most

$$\lambda_c \left(\gamma a + \gamma \ln \frac{3g(\alpha) - 1}{\gamma} + 1 + \gamma \right)$$

The above quantity is more than $c(1+\gamma)\lambda_c$ or else we would have a factor c approximation. Using the step function property, this gives us for $\alpha \geq \frac{\gamma}{(c-1)(1+\gamma)}$,

$$3g(\alpha) > 1 + \gamma e^{(c-1)(1+\frac{1}{\gamma}) - \frac{1}{\alpha}}$$

Now integrating over the interval of $0 \leq \alpha \leq 1$,

$$\begin{aligned} \int_0^1 3g(\alpha)d\alpha &\geq \int_{\frac{\gamma}{c(1+\gamma)}}^{\frac{\gamma}{(c-1)(1+\gamma)}} 3g(\alpha)d\alpha + \int_{\frac{\gamma}{(c-1)(1+\gamma)}}^{1} 3g(\alpha)d\alpha \\ &> \int_{\frac{\gamma}{c(1+\gamma)}}^{\frac{\gamma}{(c-1)(1+\gamma)}} \left(c(1+\gamma) - \frac{\gamma}{\alpha} \right) d\alpha \\ &\quad + \int_{\frac{\gamma}{(c-1)(1+\gamma)}}^{1} \left(1 + \gamma e^{(c-1)(1+\frac{1}{\gamma}) - \frac{1}{\alpha}} \right) d\alpha \\ &= \frac{\gamma}{c-1} - \gamma \ln \frac{c}{c-1} + 1 - \frac{\gamma}{(c-1)(1+\gamma)} \\ &\quad + \gamma e^{\frac{(c-1)(1+\gamma)}{\gamma}} \int_{\frac{\gamma}{(c-1)(1+\gamma)}}^{1} e^{-\frac{1}{\alpha}} d\alpha \end{aligned}$$

Call the function on the RHS as $\Gamma(c, \gamma)$. We can verify that $\Gamma(c, \gamma) > 3.0012$ for all positive γ. The guessing of the service cost would introduce a factor of $(1+\epsilon)$ in the term involving $\frac{1}{\alpha}$. Notice the for suitably small ϵ, this gives us a contradiction because $\int_0^1 3g(\alpha)d\alpha$ is at most 3 by Lemma 4.1. □

4.4 Uniform Facility Cost If the facility costs are uniform, we can change the way we round to an integral solution. The central theme of this section relies on the fact that if all the centers are of same cost, it is most profitable to open a center nearest to the demand points. We define the neighborhoods $N_\alpha(j)$, as before, and the thresholds $\tau(\alpha, j)$. Define the distance τ as before to the farthest facility in the set $N_\alpha(j)$. Let $\tau(0+, j)$ denote the distance to the closest facility supplying any flow to j, or $\tau(0+, j) = \min_i\{c_{ij} | x_{ij} > 0\}$. The quantity $g(0+)$ is defined analogous to $g(\alpha)$ as $\frac{\sum_{j\in D} d_j \tau(0+,j)}{\lambda_c}$. In each iteration pick the node with the smallest $\tau(\alpha, j) + \tau(0+, j)$.

If the demand node achieving the minimum is j', open a facility at the location closest to j' where a facility can be opened. Observe that the distance from that facility to j' is at most $\tau(0+, j')$. So all nodes v whose $N_\alpha(v)$ intersects the set $N_\alpha(j)$ would now incur a cost of $\tau(\alpha, v) + \tau(\alpha, j') + \tau(0+, j')$. By the choice of j', this is at most $2\tau(\alpha, v) + \tau(0+, v)$. Summing over all v, we get that the service cost is at most $2g(\alpha) + g(0+)$. Once again, the integral of $g(\alpha)$ is the optimal fractional service cost and at most 1. Notice that $g(0+) \le 1$, else the integral of g will certainly exceed 1.

By the same reasoning as in the proof earlier, if the distance cost exceeds $1+\gamma$, we can use a greedy improvement to lower it. We again assume that we do not achieve a factor of c.

$$\begin{aligned} 2g(\alpha) + g(0+) &\ge 1 + \gamma e^{(c-1)(1+\frac{1}{\gamma}) - \frac{1}{\alpha}} \\ &\qquad \text{for } \alpha \ge u = \tfrac{\gamma}{(c-1)(1+\gamma)} \\ &\ge 3g(0+) \text{ for all } \alpha \,. \end{aligned}$$

Integrating both sides using the first inequality over $(u, 1)$ and the second over $(0, u)$,

$$2 + g(0+) \ge 1 - u + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha + 3ug(0+)$$

Set $c = 2.225$. We have two cases depending on the value of u. If $u \ge \frac{1}{3}$ we have $\gamma \ge \frac{c-1}{4-c}$.

$$2 \ge 1 - u + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha + (3u-1)g(0+).$$

Since $(3u-1) \ge 0$ we get

$$2 \ge 1 - u + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha$$

We can verify that for all values of $u \ge \frac{1}{3}$ the RHS exceeds the LHS.

If $u < \frac{1}{3}$ we get

$$2 \ge (1-u)(1-g(0+)) + 2ug(0+) + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha$$

Since $(1-u) > 2u$ we get

$$2 \ge 2u(1-g(0+)) + 2ug(0+) + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha$$

Simplifying, we get

$$2 \ge 2u + \int_u^1 \gamma e^{\frac{1}{u} - \frac{1}{\alpha}} d\alpha$$

We can verify that for this value of c, and the related range of γ, the RHS is always greater than LHS.

5 Gap between Fractional and Integral Solution

One immediate question that comes to mind on reading the paper by Shmoys, Tardos and Aardal [14] is the issue of the "gap" between the integral and fractional solution. They show that they can round a fractional solution to an integral solution by increasing the total objective function cost by a factor of 3.16. One might ask – what is the inherent limitation of this method? What is the best constant one can hope for by this technique?

We give a family of examples for which a rounding strategy to *any* integral solution will be forced to increase the cost by a factor of about 1.463.

The set F is a set of k vertices. Each $f_i = 1$. The set D is a set of demand points, with each $d_j = 1$. We create $\binom{k}{\ell}$ demand points, one corresponding to *each* subset $S \subset F$ of size ℓ. This demand point is at a service distance c from the facilities in subset S. One should note that the distance to a facility $j \in F - S$ is exactly $3c$. (There must be a demand point that is adjacent to j and a facility in S.)

The cost of the minimum fractional solution to the linear programming formulation is at most $\frac{k}{\ell} + c \cdot \binom{k}{\ell}$. (This solution is obtained by assigning a fractional value at each center of $\frac{1}{\ell}$.) Since each demand point is adjacent to a set S of centers, it can be fractionally satisfied with cost c.

The optimal integral solution is

$$\min_{k'} k' + c\binom{k}{\ell} + 2c\binom{k-k'}{\ell}.$$

This cost is obtained by choosing k' centers from the set F, and $\binom{k-k'}{\ell}$ is the number of demand points at distance $3c$ from the set of chosen centers. Our ratio r is at least

$$\frac{\min_{k'}(k' + c(\binom{k}{\ell} + 2\binom{k-k'}{\ell}))}{\frac{k}{\ell} + c \cdot \binom{k}{\ell}}.$$

Let $\nu = \frac{\binom{k-k'}{\ell}}{\binom{k}{\ell}} = \Pi_{i=0}^{i=k'-1}(1 - \frac{\ell}{k-i})$
$\approx (1 - \frac{\ell}{k})^{k'} = e^{\frac{-\ell}{k}k'}$.

Thus $$r \geq \frac{\min_{k'}(k' + c(\binom{k}{\ell} + 2\nu\binom{k}{\ell}))}{\frac{k}{\ell} + c \cdot \binom{k}{\ell}}$$
$$= \frac{\min_{k'}(k' + c\binom{k}{\ell}(1 + 2e^{\frac{-\ell}{k}k'}))}{\frac{k}{\ell} + c \cdot \binom{k}{\ell}}.$$

Let $\beta = \frac{\ell}{k}k'$ and $xc\binom{k}{\ell} = \frac{k}{\ell}$. We obtain

$$r \geq \frac{\min_\beta(\beta xc\binom{k}{\ell} + c\binom{k}{\ell}(1 + 2e^{-\beta}))}{c\binom{k}{\ell}(1+x)}$$

$$= \min_\beta \frac{\beta x + 1 + 2e^{-\beta}}{1+x}.$$

This is minimized for $\beta = \ln \frac{2}{x}$. This yields

$$r \geq 1 + \frac{x}{1+x} \ln \frac{2}{x}.$$

This is the same function that we encountered in the proof of Theorem 3.2. The maximum value is obtained for $x = 0.463$ when we get $r \geq 1.463$.

6 Complete metric facility location

Recall that in this version of the problem $F = D = N$.

6.1 Lower bound We use the same technique as before to show a hardness of 1.278. However instead of Set Cover, we will use the minimum dominating set problem. The *Dominating set* problem asks for a smallest subset of vertices such that each vertex either belongs to the subset, or is adjacent to a vertex in the subset. Set cover reduces to this problem by an approximation preserving reduction, and it is also hard to approximate this problem within a factor $c' \ln n$ for any $c' < 1$, unless $NP \not\subseteq DTIME[n^{O(\log \log n)}]$.

Using a similar line of argument as in Subsection 3.2, we can argue that if the size of a minimum dominating set is k, choosing βk nodes (facilities) cannot dominate more than $1 - e^{-\beta}$ fraction of the nodes. The rest of the nodes would be at least at distance 2 from any facility. Choosing $f \cdot k = \gamma n$, the cost of solution is at least

$$\gamma\beta n + \left(1 - \frac{1}{e^\beta}\right) n + \frac{2}{e^\beta} n$$

Since the minimum dominating set guarantees a solution of cost at most $(1+\gamma)n$. Minimizing over β the ratio is at least

$$1 + \frac{\gamma \ln \frac{1}{\gamma}}{1+\gamma}$$

which exceeds 1.278 for $\gamma = 0.28$.

6.2 Upper Bound for uniform costs We need to modify the rounding scheme proposed in [14]. We define an LP in the same way as before. We can guess exactly how many centers the optimal solution picks. For every node j, order the facilities $\{i\}$ that serve node j in increasing order of service distance c_{ij}. For every α, define $N_\alpha(j)$ as the set of facilities closest to j and serving j such that their y_i values sum to at least α. The threshold of j, denoted by $\tau(\alpha, j)$, is defined as the service distance to the farthest facility in the set $N_\alpha(j)$.

$$d_j \int_0^1 \tau(\alpha, j) d\alpha = d_j \sum_i x_{ij} \cdot c_{ij}$$

$$= \text{ service cost of } j \text{ in fractional solution}$$

Recursively choose the unsatisfied node with the minimum $\tau(\alpha, j)$ value and place a facility there. If for any node j', $N_\alpha(j) \cap N_\alpha(j')$ is nonempty, serve j' by node j. Mark j and all such j' as satisfied.

It is easy to argue that each node j' gets served within service distance $2\tau(\alpha, j')$. Also notice that since each facility placed corresponds to disjoint $N_\alpha(j)$, the cost of facilities increase by at most $\frac{1}{\alpha}$. Thus this gives an $\left(\frac{1}{\alpha}, 2g'(\alpha)\right)$ approximation where $g'(\alpha) = \frac{\sum_j d_j \tau(\alpha, j)}{\lambda_c}$. The relation which holds on g' is

$$\int_0^1 g'(\alpha) d\alpha \leq 1$$

Now using this and the technique of the previous algorithm if there is no c approximation, we get twice the integral above is $\Gamma(c, \gamma)$. If $c = 2.104$, $\Gamma(c, \gamma)$ exceeds 2 for all γ. Hence the result follows.

Acknowledgments

We thank Neal Young and Uri Feige for discussions related to the hardness bound. We thank Randeep Bhatia, Moses Charikar, Rajeev Motwani and Yoram Sussmann for useful discussions. We would like to thank Chandra Chekuri for his comments on earlier drafts of this paper, and David Shmoys for providing us a copy of [14].

References

[1] M. L. Balinski, "On finding integer solutions to linear programs", *Proc. of the IBM Scientific Computing Symposium on Combinatorial Problems*, pages 225–248, (1966).

[2] G. Cornuejols, G. L. Nemhauser and L. A. Wolsey, "The uncapacitated facility location problem", *Discrete Location Theory* (ed: Mirchandani and Francis), pages 119–171, (1990).

[3] U. Feige, "A threshold of $\ln n$ for approximating set-cover", *28th ACM Symposium on Theory of Computing*, pages 314–318, (1996).

[4] D. S. Hochbaum, "Heuristics for the fixed cost median problem", *Mathematical Programming*, 22: 148–162, (1982).

[5] D. S. Hochbaum (Ed), *Approximation algorithms for NP-hard problems*, PWS Publishing Co., Boston, MA, 1996.

[6] M. Korupolu, C. G. Plaxton and R. Rajaraman, "Analysis of a local search heuristic for facility location problems", *9th ACM-SIAM Annual Symposium on Discrete Algorithms*, (1998).

[7] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses", *Management Science*, 9:643–666, (1963).

[8] J.-H. Lin and J. S. Vitter, "Approximation algorithms for geometric median problems", *Information Processing Letters*, 44:245–249, (1992).

[9] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems", *Journal of the ACM*, 41(5): 960–981, (1994).

[10] A. S. Manne, "Plant location under economies-of-scale-decentralization and computation", *Management Science*, 11:213–235, (1964).

[11] A. Mirzain, " Lagrangian Relaxation for the Star-Star Concentrator Location Problem", *Networks* 15:1–20, (1985).

[12] J. M. Mulvey and H. L. Crowder, " Cluster Analysis: An Application of Lagrangian Relaxation" *Management Science* 25:329–340, (1979)

[13] C. H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes " *Journal of Computer and System Sciences* 43:425-440, (1991).

[14] D. Shmoys, E. Tardos and K. Aardal, "Approximation algorithms for facility location problems", *29th ACM Symposium on Theory of Computing*, pages 265–274, (1997).

[15] J. F. Stollsteimer, "A working model for plant numbers and locations", *J. Farm Econom.*, 45: 631–645, (1963).