The Facility Location Problem with Concave Cost Functions

MohammadTaghi Hajiaghayi, Mohammad Mahdian, Vahab S. Mirrokni

Laboratory for Computer Science, Massachusetts Institute of Technology.

Abstract. In this paper we define and study a generalized version of facility location problem in which facility cost functions depend on the number of clients assigned to the facility. There is an associated cost function for each facility that depends on the number of clients assigned to it. We focus on the case of concave facility cost functions, and present greedy 1.94 and 1.52 approximation algorithms for this case. We will also consider various generalizations and variants of the problem and give an $O(\log n)$ approximation algorithm for the non-metric generalized facility location problem.

1 Introduction

The facility location problem is a central problem in operation research. In this problem, we have a set of clients and a set of facilities, and we want to connect each client to a facility in a way that minimizes the total cost. There are two types of costs associated with a solution: the connection costs and the facility costs. The *connection cost* between each client j and facility i is a number c_{ij} that is given. We will have to pay this amount if we want to connect client j to facility i. Each facility i also has a *facility cost* f_i, that is the amount we have to pay if we decide to connect at least one client to it. The cost of a solution is the summation of the connection costs and the facility costs associated with the solution. We will generally assume that the connection costs obey the metric inequality.

The facility location problem is well-studied in the field of approximation algorithms, and a number of different approximation algorithms have been proposed for this problem using a variety of techniques [11, 14, 6, 7, 10, 8, 1–5, 8, 9, 13].

A natural generalization of the facility location problem is to allow the facility cost of a facility to be an arbitrary concave function of the number of cities connected to it. This question was first asked by Tom Leighton and was motivated by its applications in placing servers on the Internet. The reason it is assumed that the facility cost function is concave is the following principle of economics: As the number of clients increases, the cost per client will decrease, since they share some common expenses.

The main result of this paper is a 1.94 approximation algorithm for the concave facility location problem. Our algorithm is a natural generalization of an algorithm by Jain et al. [7]. The analysis uses the techniques of dual-fitting and factor-revealing LPs introduced in [6]. We will also consider more general versions of the problem by relaxing the conditions of concavity for facility costs and metricity for connection costs.

This paper is organized as follows. In section 2, we give a formal definition of the problem, and observe that it can be reduced to the capacitated facility location problem with hard capacities. In section 3, we present a greedy 1.94 approximation algorithm for the concave facility location problem. In section 4, we prove the approximation factor 1.94 for our greedy algorithm. In sections 5 and 6, we study generalizations and variants of the problem such as allowing the facility cost function to be slightly non-concave, facility location with convex cost functions, and relaxing the metric inequality for the connection costs.

2 The problem

The *generalized facility location* problem is define as follows. We are given a set $\mathcal{F}$ of *facilities* (a.k.a servers); a set $\mathcal{D}$ of *clients* (a.k.a. cities or demands); a *facility cost function* $f_i : N \to N$ for every facility $i \in \mathcal{F}$, which specifies the cost of the facility as a function of the number of clients served by it; and finally *connection costs* c_{ij} between facility i and client j. We generally assume that the connection costs are *metric* (i.e., is symmetric and satisfies the triangle inequality), unless otherwise stated. The objective of the problem is to find an assignment ψ of *all* clients to facilities (i.e., $\psi : \mathcal{D} \mapsto \mathcal{F}$) with minimum total cost. The total cost is the sum of connection costs ($\sum_{j\in\mathcal{D}} c_{\psi(j),j}$) and facility costs ($\sum_{i\in\mathcal{F}} f_i(|\{j \in \mathcal{D} : \psi(j) = i\}|)$).

The focus of this paper is on the *concave facility location problem*, i.e., the generalized facility location problem with a guarantee that all facility cost functions are concave. Recall that a function $f : N \to N$ is called concave if and only if for each $x \geq 1$, $f(x+1) - f(x) \leq f(x) - f(x-1)$.

We observe that the concave facility location problem can be reduced to the *capacitated facility location with hard capacities*. In the capacitated facility location problem, each facility i also has a capacity u_i, which is the maximum number of clients that can be served by this facility. The problem has two variants; in the first one, the capacities are *hard* in that facility i can be opened at most once to serve at most u_i demands; in the second variant the capacities are *soft*; that is, facility i may be opened k times to serve up to ku_i demands at a cost of kf_i. It is easy to observe that both of these problems are special cases of the generalized facility location problem. The capacitated facility location problem with soft capacities can be reduced to the uncapacitated facility location problem [11], and the best known approximation algorithm for this problem [11] is obtained via this reduction. However, hard capacities are known to be problematic for techniques that rely on linear programs (such as LP-rounding and primal-dual algorithms). The main problem is that natural linear program has a large integrality gap for the case of hard capacities. Recently, Pal et al. [12] used local search techniques to give the first constant factor approximation algorithm for this problem, achieving a factor of $9 + \epsilon$. However, the running time of this algorithm is prohibitive for applications with large data sets.

Proposition 1. *Assume there is a polynomial time α-approximation algorithm for the capacitated facility location problem with hard capacities. Then the concave facility location problem can be approximated within a factor of α in polynomial time.*

Proof. For each facility $i \in \mathcal{F}$, we place n copies of this facility, where the j'th copy has opening cost $f_i(j)$ and capacity j. We use the α-approximation algorithm for the capacitated facility location problem to solve this instance. Since the facility cost function is concave, we can assume without loss of generality that at most one of these n copies is opened in this solution. Therefore, the solution to the capacitated facility location instance can be easily transformed to a solution for the concave facility locatino problem with the same total cost.

The above proposition together with the algorithm of Pal et al. [12] gives a $9+\epsilon$-approximation algorithm for the concave facility location problem. In the next section, we will show a greedy algorithm for this problem achieving a factor of 1.94.

3 The Greedy Algorithm

In this section, we present the main result of this paper that is a greedy 1.94-approximation algorithm for the concave facility location problem. The approximation factor of this algorithm is much better than the algorithm mentioned in the proof of Theorem 1. In addition, since the previous algorithm uses the approximation algorithm of the capacitated facility location problem with hard capacities, and the current algorithm for this problem uses the local search method, its running time is very high, especially because we reduce an instance of concave facility location with $O(n)$ facilities to an instance of capacitated facility location with $O(n^2)$ facilities. Our method is an extension of the method used by Jain et al. [7] and Mahdian et al. [10]. First, we define stars.

Definition 1. *A star consists of one facility and several clients. For star S, consisting of clients $a_1, a_2, \ldots, a_k$ and facility p, cost of S, denoted by $c(S)$ or c_S, is defined as $\sum_{i=1}^{k} c_{p,a_i} + f_p(k)$ i.e., the sum of the connection costs of clients to the facility p plus the facility cost function of p for k clients.*

Suppose an algorithm finds a solution of cost θ to the concave facility location problem, and also it finds values α_j for every client $j \in \mathcal{C}$ as the contribution of client j in the total cost. In addition, assume

$\sum_{j\in\mathcal{C}} \alpha_j = \theta$ and there is a constant $\gamma \geq 1$ such that for every star S, $\sum_{j\in S\cap\mathcal{C}} \alpha_j \leq \gamma c_S$. We now consider an optimal solution OPT to the problem. Let i be a facility that is opened in OPT. For set D of clients that are connected to facility i in OPT, we can write $\sum_{j\in D} \alpha_j \leq \gamma(f_i + \sum_{j\in D} c_{ij})$ or $\sum_{j\in D} \alpha_j \leq \gamma c_S$ where S is the star consisting of i and D. By summing up the inequalities for every star that is picked in OPT, we obtain $\theta = \sum_{j\in\mathcal{C}} \alpha_i \leq \gamma \sum_{S\in\mathrm{OPT}} c_S = \gamma \cdot \mathrm{cost}(\mathrm{OPT})$. Therefore if we can find such an algorithm with a constant γ, γ is the approximation factor of the algorithm.

It is worth mentioning that this approach can also be considered using LP-duality. The problem can be formalized by an integer linear program based on stars, and the α_i's are the variables of the dual program in which we relax inequalities by a constant factor γ. The reader is referred to Jain et al. [7] to see this method called the dual-fitting method in more details.

In the next section using the approach discussed above, we show that simple greedy **Algorithm A** presented below is a 1.94 approximation algorithm for the concave facility location problem. In this algorithm, we use a notion of time (lines 1 and 6), such that every event can be associated with the time at which it happened. Also each client j has a budget from which it can offer some money to facilities; if j is unconnected and its budget is more than the cost of the connection to a facility i, it offers the extra budget to i (line 9); and if j is connected to a facility i', it offers to a facility i the amount by which it can save by switching its facility from i' to i (line 10). We note that at any time, the budget of each connected client is equal to its current connection cost plus its total contribution toward open facilities.

Algorithm A: greedy algorithm for concave facility location
Input: Metric connection costs c_{ij} for each facility i and client j.
Concave facility cost functions $f_i : N \rightarrow N$ for each facility i.
Output: For each client j, a facility $p(j)$ to which j is assigned.
For each client j, contribution of client j to the total cost (α_j).
begin
1 **let** $t = 0$
2 **for** each facility i **let** $\mathrm{level}_p = 0$
3 **for** each client j
4 **let** $p(j) =$ **null**
5 **let** $\mathrm{budget}(j) = 0$
6 **while** there is an unconnected client increase time t
7 **for** each unconnected client j **let** $\mathrm{budget}(j) = t$
8 **for** each client j
9 **if** $p(j) =$ **null let** $\mathrm{offer}(j,i) = \max(\mathrm{budget}(j) - c_{ij}, 0)$
10 **else let** $\mathrm{offer}(j,i) = \max(c_{p(j)j} - c_{ij}, 0)$
11 **while** there is a facility i and $k - \mathrm{level}_i$ clients $a_1, \cdots, a_{k-\mathrm{level}_i}$ $(k > \mathrm{level}_i)$
which contains at least one unconnected client and $\sum_{j=1}^{k-\mathrm{level}_i} \mathrm{offer}(a_j, i) = f_i(k) - f_i(\mathrm{level}_i)$
12 **let** $\mathrm{level}_i = k$
13 **for** each $1 \leq j \leq k - \mathrm{level}_i$ **let** $p(a_j) = i$
14 **for** each client j
15 **let** $\alpha(j) = \mathrm{budget(j)}$, the time that j first gets connected
end

The proof of the following lemma is clear from the algorithm and the discussion above.

Lemma 1. *The total cost of the solution found by* **Algorithm A** *is equal to the sum of* α_j*'s.*

The above algorithm is similar to the greedy algorithm of Jain et al. [7]. The difference is that here we define the concept of level for facilities that is the number of clients assigned to it and the events in which we assign clients to facilities depend on the level of vertices.

4 The approximation factor

In this section, we show that **Algorithm A** is indeed a 1.94-approximation algorithm for the concave facility location problem. We prove this by showing that for each star S, the ratio of the sum of α_j's of all clients contained in S to the total cost of S is at most $\gamma \approx 1.94$. Analogous to the work of Jain et al. [7], our approach is as follows. First, based on the behavior of the algorithm we obtain some linear constraints called *factor revealing-LP* on α_i's and the cost of S. Next, we show that for any feasible solution of the LP (not necessarily for the one obtained from the algorithm) our objective ratio is at most $\gamma \approx 1.94$. Here, an LP-solver helps us to guess such a ratio, and then using complicated calculations we prove this upper bound.

To derive the factor-revealing LP, first we need some definitions and notations. Consider a star S consisting of a facility p and k clients numbered 1 through k. Let d_j denote the connection cost between facility p and client j, and α_j denote the share of j of the total expenses (see the definition of α in the algorithm). The cost of the star is $f_p(k) + \sum_{i=1}^{k} d_i$. For simplicity, we set $f = f_p(k)$. Without loss of generality, we assume $\alpha_1 \le \alpha_2 \le \cdots \le \alpha_k$. Let the *critical time* for a client i be the time just before i gets connected for the first time, i.e., when $t = \alpha_i - \epsilon$ where ϵ is very small. At the critical time for client i, each of the clients $1, 2, \ldots, i-1$ might be connected to a facility. For every $j < i$, if client j is connected to some facility at time t, let $r_{j,i}$ denote the connection cost between this facility and client j; otherwise, let $r_{j,i} := \alpha_j$ (in this case $\alpha_i = \alpha_j$).

First we note that since the budget of a client remains constant when it gets connected to a facility, and it may not get connected to another facility with a higher connection cost, $r_{j,j+1} \ge r_{j,j+2} \ge \cdots \ge r_{j,k}$. Now we obtain more constraints.

Lemma 2. *At the critical time for a client i, for every subset $S_1 \subseteq \{1, \ldots, i-1\}$ and every subset $S_2 \subseteq \{i, \ldots, k\}$ ($S_2 \neq \emptyset$),*

$$\sum_{j \in S_1} \max(r_{j,i} - d_j, 0) + \sum_{j \in S_2} \max(\alpha_i - d_j, 0) \le f_p(|S_1| + |S_2|). \tag{1}$$

In particular, $\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f_p(k) = f$.

Proof. The amount client j offers to facility f at time $t = \alpha_i - \epsilon$ is $\max(r_{j,i} - d_j, 0)$ if $j < i$, and $\max(t - d_j, 0)$ if $j \ge i$. By the definition of $r_{j,i}$ this holds even if $j < i$ and $\alpha_i = \alpha_j$. From the algorithm, the total offer of clients in $S_1 \cup S_2$ to facility p may not become larger than the cost of facility p at level $|S_1| + |S_2|$, since otherwise all these clients were assigned to the facility p. Thus, for all i, $\sum_{j \in S_1} \max(r_{j,i} - d_j, 0) + \sum_{j \in S_2} \max(\alpha_i - d_j, 0) \le f_p(|S_1| + |S_2|)$. In particular by setting $S_1 = \{1, \ldots, i-1\}$ and $S_2 = \{i, \ldots, k\}$, $\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f$. □

So far, we have not used the triangle inequality of connection costs and concavity of facility cost functions. We use these assumptions in the next lemma.

Lemma 3. *At the critical time for a client i, for all clients j such that $1 \le j < i$,*

$$\alpha_i \le \alpha_j + r_{j,i} + d_i + d_j. \tag{2}$$

Proof. Let p' be the facility that j is connected to at time $t = \alpha_i - \epsilon$. By the triangle inequality and the definition of $r_{j,i}$, the connection cost $c_{p'i}$ between client i and facility p' is at most $c_{p'j} + d_i + d_j$. It is not hard to see that by the definition of $r_{j,i}$, we have $c_{p'j} \le r_{j,i}$. Thus these two inequalities imply $c_{p'i} \le r_{j,i} + d_i + d_j$.

Furthermore, if the level of p' is equal to l when client j gets connected to it, $c_{p'i}$ can not be less than $t + f_{p'}(l+1) - f_{p'}(l)$, since otherwise the client i should be connected to the facility p' at a time earlier than t, which is a contradiction. This shows that: $\alpha_i - \epsilon = t < c_{p'i} + f_{p'}(l+1) - f_{p'}(l)$. From the last inequality and this one, $\alpha_i \le r_{j,i} + d_i + d_j + f_{p'}(l+1) - f_{p'}(l)$. From the fact that $f_{p'}$ is a concave function, it turns out that for all $q < l$, $f_{p'}(l+1) - f_{p'}(l) \le \frac{f_{p'}(l) - f_{p'}(q)}{l-q}$. Now consider the time α_j at which j gets connected to facility p'. Let q be the level of p' before time α_j and at this time $l-q$ new clients, $b_1, b_2, \ldots, b_{l-q}$ are assigned to facility p'. At time α_j the total amount of these $l-q$ clients' offer to p' is equal to $f_{p'}(l) - f_{p'}(q)$. The amount of b_i's offer to facility p' is equal to either $\alpha_j - c_{b_ip'}$ or $c_{b_ip''} - c_{b_ip'}$ depends on the situation of client b_i at time α_j (whether it was assigned to a facility p'' or not). In either case, b_i's offer is less than or equal to $\alpha_j - c_{b_i p'}$, thus

$$f_{p'}(l) - f_{p'}(q) = \sum_{i=1}^{l-q} \text{offer}(b_i) \le \sum_{i=1}^{l-q} \alpha_j = (l-q)\alpha_j \Rightarrow \frac{f_{p'}(l) - f_{p'}(q)}{l-q} \le \alpha_j$$

Combining all these inequalities together, we get the following inequality. For every $1 \le j < i \le k$, $\alpha_i \le r_{j,i} + d_i + d_j + \alpha_j$. □

The following optimization program, called the factor-revealing LP, can be obtained from the above inequalities. We note that by scaling $f + \sum_{i=1}^{k} d_i = 1$ and introducing new variables and new constraints for function max we can obtain a linear program.

$$\begin{array}{ll}
\text{maximize} & \dfrac{\sum_{i=1}^{k} \alpha_i}{f + \sum_{i=1}^{k} d_i} \\
\text{subject to} & \forall\, 1 \le i < k: \ \alpha_i \le \alpha_{i+1} \\
& \forall\, 1 \le j < i < k: \ r_{j,i} \ge r_{j,i+1} \\
& \forall\, 1 \le j < i \le k: \ \alpha_i \le \alpha_j + r_{j,i} + d_i + d_j \\
& \forall\, 1 \le i \le k, S_1 \subseteq \{1, \ldots, i-1\}, S_2 \subseteq \{i, \ldots, k\}: \ \sum_{j \in S_1} \max(r_{j,i} - d_j, 0) \\
& \qquad\qquad + \sum_{j \in S_2} \max(\alpha_i - d_j, 0) \le f_p(|S_1| + |S_2|) \\
& \forall\, 1 \le j \le i \le k: \ \alpha_j, d_j, f, r_{j,i} \ge 0
\end{array} \tag{3}$$

The size of the above program is large (exponential) because of the forth set of inequalities and it is hard to find out the solution of the problem for large k's. In order to solve this problem, we observed that using Lemma 2, we can relax the forth set of inequalities and still get the approximation factor 1.94.

$$\begin{array}{ll}
\text{maximize} & \dfrac{\sum_{i=1}^{k} \alpha_i}{f + \sum_{i=1}^{k} d_i} \\
\text{subject to} & \forall\, 1 \le i < k: \ \alpha_i \le \alpha_{i+1} \\
& \forall\, 1 \le j < i < k: \ r_{j,i} \ge r_{j,i+1} \\
& \forall\, 1 \le j < i \le k: \ \alpha_i \le \alpha_j + r_{j,i} + d_i + d_j \\
& \forall\, 1 \le i \le k: \ \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f \\
& \forall\, 1 \le j \le i \le k: \ \alpha_j, d_j, f, r_{j,i} \ge 0
\end{array} \tag{4}$$

After this relaxation, the number of inequalities in the optimization program is polynomial.

Theorem 1. *Let γ be $\sup_k\{z_k\}$, where z_k is the solution of the factor-revealing LP. Then* **Algorithm A** *gives a γ-approximation algorithm for the concave facility location problem.*

Proof. Since the values α_i, d_j, f and r_{ji} obtained from **Algorithm A** satisfy all inequalities in the LP, the values of the objective function for them is at most z_k. It implies for each star S consisting of one facility and k clients $i_1, \cdots, i_k$, $\sum_{j=1}^{k} \alpha_{i_j}$ is at most $z_k c_S$. The proof of the lemma follows from this fact and Lemma 1. □

In the next step, we use an LP-solver like CPLEX to obtain the optimum solution of the factor revealing LP for fixed k. The results are shown in Table 1.

k	$\max_{i\leq k} z_i$
10	1.83517
20	1.88389
50	1.91573
100	1.92652
200	1.93193

Table 1. Solutions of the factor-revealing LP

Using these experimental results one can observe that $\gamma \approx 1.94$; however all of them are just lower bounds for the LP. To obtain the desired approximation ratio, we need to prove an upper bound 1.94 on the maximum solution of the LP. The proof needs so much calculations, and thus it is presented in Appendix A. It is worth mentioning Mahdian et al. [7, 11] also use these kinds of tedious calculations for the facility location problem; however since our LP is different from theirs, the calculations are different. Finally we have the following theorem:

Theorem 2. **Algorithm A** *is a 1.94-approximation algorithm with running time $O(n^4)$ for the concave facility location problem, where $n = \max(n_f, n_c)$.*

The following improvement for the concave facility location problem has been suggested by one of the referees of SODA 2003. It is worth mentioning, still we use **Algorithm A** for more general connection costs in Section 6.

Theorem 3. *There exists a 1.52-approximation algorithm for the concave facility location problem.*

Proof. The main idea is that in our problem any concave function $f(x)$ can be represented by $min_k\{(f(k)-f(k-1))x+kf(k-1)-(k-1)f(k)\}$, $1 \leq k \leq n$. Now, we take each facility i with concave cost function f_i, and replace it by multiple facilities $i_1, i_2, \cdots, i_n$ such that facility i_k has opening cost $kf_i(k-1)-(k-1)f_i(k)$. In addition, each unit of demand routed to this location costs $f_i(k) - f_i(k-1)$ extra at the facility. This cost $f_i(k) - f_i(k-1)$ can be added to the distance metric. For this facility location problem, Mahdian et al. [11] have a 1.52 approximation.

5 Generalizations

In this section, we consider more general variants of the problem such as the problem with relaxed metric inequality.

5.1 Other facility cost functions

So far, we have shown approximation algorithms for the concave facility location problem. In this section, we consider the generalized facility location problem with more general facility cost functions. First, we give a polynomial time algorithm for the generalized facility location problem with convex facility cost functions. Recall that a function f is called *convex* if for every $x \geq 1$, $f(x+1) - f(x) \geq f(x) - f(x-1)$.

Theorem 4. *The generalized facility location problem with convex facility cost functions can be solved in polynomial time.*

Proof. First we reduce the problem to the capacitated facility location problem with unit hard capacities. Next, we show how this problem can be solved in polynomial time. For each facility $i \in \mathcal{F}$, we place n copies of unit-capacity facilities where f_i^j, the opening cost of the jth facility of the ones corresponding to facility $i \in \mathcal{F}$, is $f_i(j+1) - f_i(j)$, $0 \leq j \leq n-1$. The correctness of the reduction follows from the fact that if we use a facility f_i^j, we should use all facilities f_i^k, $k \leq j$, since function f_i is convex. We now solve the problem by minimum weighted matching on bipartite graphs. We construct a bipartite graph $G = (X \cup Y, E)$ as follows. For each client j, we place a vertex in the set X, for each facility i, we place a vertex in the set Y, and finally we place an edge $\{j, i\}$ in E between a client j and a facility i with weight $c_{ij} + f_i$. We can easily observe by solving minimum weighted matching [15] for G, one can find an optimum assignment for the original problem. □

The problem of finding a constant factor approximation algorithm for general facility cost functions is open. Here, we observe that **Algorithm A** can be used to find constant factor approximation algorithms for cost functions that are close to concave functions.

Definition 2. *A function $f : N \to N$ is a* c-close concave function *if there exists a concave function g such that $\forall x \in N : \frac{g(x)}{c} \leq f(x) \leq g(x)$. The function $f : N \to N$ is* c-concave *if and only if for all l and q such that $q < l$, we have $f(l+1) - f(l) \leq c\frac{f(l)-f(q)}{l-q}$.*

In the case of c-close concave functions, we have the following simple theorem.

Theorem 5. *There is a constant factor approximation algorithm of factor $1.94c$ for the generalized facility location problem with cost functions that are c-close to concave functions.*

Proof. Consider a function g_i such that $\frac{g_i}{c} \leq f_i \leq g_i$. We use **Algorithm A** to solve the problem for facility cost functions g_i's. We know that the cost of this solution is at most 1.94 times the optimal solution for facility cost functions g_i's. Using the inequality $\frac{g_i}{c} \leq f_i \leq g_i$, the optimal solution for g_i's is at most c times the optimal solution for f_i's. Thus, the approximation factor is $1.94c$.

For c-concave functions, one can observe that by a similar proof of Lemma 3, we can prove that if f_i's are c-concave, then for all $1 \leq j < i \leq k$, $\alpha_i \leq c\alpha_j + r_{j,i} + d_i + d_j$. Therefore, using **Algorithm A** for functions f_i's, the approximation factor of the algorithm is the optimal solution of the same factor revealing LP except the third set of inequalities which are replaced by $\alpha_i \leq c\alpha_j + r_{j,i} + d_i + d_j$. Table 2 shows a summary of the results obtained by solving the factor-revealing LP using CPLEX for $k = 100$. From the experimental results, it turns out that z_k depends on c by an asymptotically linear function and the approximation factor is again a constant.

c	$\max_{i \leq k} z_i$	λ	$\max_{i \leq k} z_i$
0.2	1.6579	0.2	1.3348
0.5	1.75595	0.5	1.6102
1	1.92652	2	2.4456
2	2.29422	10	4.4621
10	5.98046	50	5.0569

Table 2. Approximation factor for c-concave functions and λ-parameterized metric($k = 100$)

5.2 More general connection costs

It is easy to see that the facility location problem (and therefore the generalized faciltiy location problem) is NP-hard to approximate within a factor less than $O(\ln n)$ if the connection costs are not metric. Also, it is not difficult to see that the classical set cover algorithm can approximate the non-metric concave facility location problem within a factor of $O(\ln n)$. However, one can observe that **Algorithm A** works very well when the metric inequality is somewhat relaxed. In this case, instead of the triangle inequality ($AC \leq AB + BC$) a parameterized triangle inequality ($AC \leq \lambda(AB + BC)$) is satisfied. It is straightforward to restate the proof of Lemma 3 and prove that for all $1 \leq j < i \leq k$, $\alpha_i \leq \lambda(r_{ji} + d_i + d_j) + \alpha_j$. Therefore, we have the same factor-revealing LP except the third set of inequalities which are replaced by $\alpha_i \leq \lambda(r_{ji} + d_i + d_j) + \alpha_j$. Using CPLEX, we obtained the optimum solution of this factor-revealing LP for $k = 100$. Table 2 shows the empirical results for different values of λ. These results show that Algorithm A works well when the metric inequality is somewhat relaxed.

6 General connection costs

It is easy to see that the facility location problem (and therefore the generalized faciltiy location problem) is NP-hard to approximate within a factor less than $O(\ln n)$ if the connection costs are not metric. Also, it is not difficult to see that the classical set cover algorithm can approximate the non-metric concave facility location problem within a factor of $O(\ln n)$. However, this reduction does not work for the non-metric *generalized* facility location problem. For this case, we can prove that Algorithm A has an approximation factor of $\ln n$, where n is the number of clients.

Theorem 6. *Algorithm A achieves an approximation factor of* $\ln n$ *for the generalized facility location problem when the connection cost is non-metric.*

Proof. Recall the definitions of α_j, d_j, and f in Section 4. By Lemma 2, we have $\sum_{j=i}^{k}(\alpha_i - d_j) \leq f$ (Notice that the concavity assumption was not used in the proof of Lemma 2). Thus,

$$\alpha_i \leq \frac{1}{k-i+1}(f + \sum_{j=i}^{k} d_j) \leq \frac{1}{k-i+1}(f + \sum_{j=1}^{k} d_j).$$

It follows that

$$\sum_{i=1}^{k} \alpha_i \leq \sum_{i=1}^{k} \frac{1}{k-i+1}(f + \sum_{j=1}^{k} d_j) = H_k(f + \sum_{j=1}^{k} d_j) \leq (\ln n) c_S.$$

The above theorem implies that the capacitated facility location problem can be approximated by a factor of $\ln n$. To the best of our knowledge, this is the first approximation algorithm for the (hard) capacitated facility location problem when the connection cost is non-metric. Also, it is not difficult to observe that if instead of the metric inequality, connection costs satisfy a relaxed version of the metric inequality, then by proving a relaxed version of the inequality in Lemma 3 and solving the corresponding factor-revealing LP, one can obtain the approximation factor of the algorithm.

Acknowledgments. We would like to thank Tom Leighton, Rajmohan Rajaraman and Ravi Sundaram for introducing the problem.

References

1. V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, 2001.
2. M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, October 1999.
3. F.A. Chudak. Improved approximation algorithms for uncapacited facility location. In R.E. Bixby, E.A. Boyd, and R.Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture Notes in Computer Science*, pages 180–194. Springer, Berlin, 1998.
4. F.A. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. unpublished manuscript, 1998.
5. S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31:228–248, 1999.
6. K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. Approximation algorithms for facility location via dual fitting with factor-revealing lp. unpublished.
7. K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problem. In *Proceedings of 34rd ACM Symposium on Theory of Computing*, 2002.
8. K. Jain and V.V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13, October 1999.

9. M.R. Korupolu, C.G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1–10, January 1998.
10. M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. A greedy facility location algorithm analyzed using dual fitting. In *Proceedings of 5th International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 2129 of *Lecture Notes in Computer Science*, pages 127–137. Springer-Verlag, 2001.
11. M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. to appear in APPROX 2002, 2002.
12. Martin Pal, Eva Tardos, and Tom Wexler. Facility location with nonuniform hard capacities. In *Proceedings of the 42th Annual IEEE Symposium on Foundations of Computer Science*, 2001.
13. D.B. Shmoys, E. 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.
14. M. Sviridenko. An 1.582-approximation algorithm for the metric uncapacitated facility location problem. Ninth Conference on Integer Programming and Combinatorial Optimization, 2002.
15. Douglas B. West. *Introduction to Graph Theory.* Prentice Hall Inc., Upper Saddle River, NJ, 1996.

A Proof of Theorem 3

In order to prove this upper bound, first we prove the following lemma which allows us to prove the theorem on the case of sufficiently large k. The proof of the following lemma is the same as the proof of Lemma 14 in [7] and hence omitted.

Lemma 4. *If z_k denotes the solution to the factor-revealing LP, then for every k, $z_k \le z_{2k}$.*

Now, in order to prove the approximation factor, the objective is to combine the inequalities of the Program 4 to derive an inequality of the form $\sum_{i=1}^{k} \alpha_i \le \gamma(f + \sum_{i=1}^{k} d_i)$. Such a γ will be an upper bound on the solution of the Program 4.

We start by relaxing the forth inequality of the Program 4 to the following inequality

$$\sum_{j=i}^{l_i}(\alpha_i - d_j) + \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \le f, \tag{5}$$

where as in [6] we define l_i as follows

$$l_i = \begin{cases} p_2 k & \text{if } i \le p_1 k \\ k & \text{if } i > p_1 k \end{cases} \tag{6}$$

Here p_1 and p_2 are constants with $0 < p_1 < p_2 < 1$ which will be fixed later in the proof.

Inequality 5 implies

$$\alpha_i \le \frac{1}{l_i - i + 1}\left(f + \sum_{j=i}^{l_i} d_j - \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0)\right). \tag{7}$$

We multiply both sides of the above inequality by a constant θ_i that will be fixed later, and add the resulting inequalities. This will imply the following inequality.

$$\sum_{i=1}^{k} \theta_i \alpha_i \le \sum_{i=1}^{k} \frac{\theta_i}{l_i - i + 1}\left(f + \sum_{j=i}^{l_i} d_j - \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0)\right) \tag{8}$$

Let ζ and λ_j be defined as follows.

$$\zeta = \sum_{i=1}^{k} \frac{\theta_i}{l_i - i + 1} \tag{9}$$

$$\lambda_j = \begin{cases} \sum_{i=1}^{j} \frac{\theta_i}{l_i - i + 1} & \text{if } j \le p_2 k \\ \sum_{i=p_1 k+1}^{p_2 k} \frac{\theta_i}{l_i - i+1} & \text{if } j > p_2 k \end{cases} \tag{10}$$

Therefore, Inequality 8 can be written as follows.

$$\sum_{i=1}^{k} \theta_i \alpha_i \le \zeta f + \sum_{i=1}^{k} \lambda_i d_i - \sum_{i=1}^{k} \frac{\theta_i}{l_i - i + 1} \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \tag{11}$$

In the above inequality, some θ_i's are greater than 1 and others are less than or equal to 1. The next step is to use the inequality $\alpha_i \le \alpha_j + r_{j,i} + d_i + d_j$ to make the coefficient of all α_i's on the left-hand side of the inequality equal to 1. We assume θ_i's are chosen in a way that

$$\sum_{i=1}^{k} \theta_i = k. \tag{12}$$

Also, we assume there is a constant p_3 such that

$$\forall i \le p_3 k,\ \theta_i > 1 \qquad \text{and} \qquad \forall i > p_3 k,\ \theta_i \le 1 \tag{13}$$

We will make sure that θ_i's satisfy the above constraints when we fix their values later in the proof. Now consider the inequality $\alpha_i \le \alpha_j + r_{j,i} + d_i + d_j$ for $i > p_3 k$ and $j \le p_3 k$. Multiply both sides of this inequality by a constant $\omega_{i,j} \ge 0$, and add up all these inequalities with Inequality 11.
If we choose ω_{ij}'s in such a way that

$$\sum_{i > p_3 k} \omega_{i,j} = \theta_j - 1 \qquad \forall j \le p_3 k \qquad \text{and} \qquad \sum_{j \le p_3 k} \omega_{i,j} = 1 - \theta_i \qquad \forall i > p_3 k \tag{14}$$

we will get the following inequality:

$$\sum_{i=1}^{k} \alpha_i \le \zeta f + \sum_{i=1}^{k} (\lambda_i + |1 - \theta_i|) d_i + \sum_{i > p_3 k} \sum_{j \le p_3 k} \omega_{i,j} r_{j,i} - \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{\theta_i}{l_i - i + 1} \max(r_{j,i} - d_j, 0) \tag{15}$$

Now, we define $p_4 < p_1$ so that we only use triangle inequalities $\alpha_i \le r_{ji} + \alpha_j + d_i + d_j$ for $i > p_2 k$ and $p_4 k < j \le p_3 k$ or for $p_3 k < i \le p_2 k$ and $j \le p_4 k$. From this definition of p_4, first we need that

$$\sum_{j=p_3 k+1}^{p_2 k} (1 - \theta_j) \le \sum_{j=1}^{p_4 k} (\theta_j - 1) \tag{16}$$

and also we choose ω_{ij}'s such that $0 < p_4 \le p_1$ and

$$\omega_{ij} \ne 0 \Rightarrow (i > p_2 k \text{ and } p_4 k < j \le p_3 k) \text{ or } (p_3 k < i \le p_2 k \text{ and } j \le p_4 k) \tag{17}$$

Now, using the fact that $r_{ji} \ge r_{ji+1}$, we can write Inequality 15 as follows:

$$\sum_{i=1}^{k} \alpha_i - \zeta f \le \sum_{i=1}^{k} (\lambda_i + |1-\theta_i|) d_i + \sum_{i=p_2 k+1}^{k} \sum_{j=p_4 k+1}^{p_3 k} \omega_{i,j} r_{j,i} + \sum_{i=p_3 k+1}^{p_2 k} \sum_{j \le p_4 k} \omega_{i,j} r_{j,i} - \sum_{j=1}^{k} \sum_{i=j+1}^{k} \frac{\theta_i}{l_i - i + 1} \max(r_{j,i} - d_j, 0)$$

$$\le \sum_{i=1}^{k} (\lambda_i + |1-\theta_i|) d_i + \sum_{j=p_4 k+1}^{p_3 k} r_{j,p_2 k} \Big(\sum_{i=p_2 k+1}^{k} \omega_{i,j} \Big) + \sum_{j=1}^{p_4 k} r_{j,p_3 k} \Big(\sum_{i=p_3 k+1}^{p_2 k} \omega_{i,j} \Big) - \sum_{j=1}^{p_3 k} \sum_{i=j+1}^{k} \frac{\theta_i}{l_i - i + 1} \max(r_{j,i} - d_j, 0)$$

Using Equations 14 and 17, it turns out that $\sum_{i > p_2 k} \omega_{i,j} = \theta_j - 1$ for $j \le p_4 k$ and $\sum_{p_3 k < i \le p_2 k} \omega_{i,j} = \theta_j - 1$ for $p_4 k < j \le p_3 k$. Thus,

$$\sum_{i=1}^{k}\alpha_i-\zeta f\le\sum_{i=1}^{k}(\lambda_i+|1-\theta_i|)d_i+\sum_{j=p_4k+1}^{p_3k}r_{j,p_2k}(\theta_j-1)+\sum_{j=1}^{p_4k}r_{j,p_3k}(\theta_j-1)$$
$$-\sum_{j=1}^{p_4k}\sum_{i=j+1}^{k}\frac{\theta_i}{l_i-i+1}\max(r_{j,i}-d_j,0)-\sum_{j=p_4k+1}^{p_3k}\sum_{i=j+1}^{k}\frac{\theta_i}{l_i-i+1}\max(r_{j,i}-d_j,0)$$

Notice that for $i\ge p_2k$, $r_{j,i}\le r_{j,p_2k}$ and for $i\ge p_3k$, $r_{j,i}\le r_{j,p_3k}$. After substitution of these values in the above inequality, we get the following:

$$\begin{aligned}\sum_{i=1}^{k}\alpha_i-\zeta f&\le\sum_{i=1}^{k}(\lambda_i+|1-\theta_i|)d_i+\sum_{j=p_4k+1}^{p_3k}r_{j,p_2k}(\theta_j-1)+\sum_{j=1}^{p_4k}r_{j,p_3k}(\theta_j-1)\\&\quad-\sum_{j=1}^{p_4k}\sum_{i=j+1}^{p_3k}\frac{\theta_i}{l_i-i+1}(r_{j,p_3k}-d_j)-\sum_{j=p_4k+1}^{p_3k}\sum_{i=j+1}^{p_2k}\frac{\theta_i}{l_i-i+1}(r_{j,p_2k}-d_j)\\&\le\sum_{i=1}^{p_3k}(\lambda_i+2(\theta_i-1))d_i+\sum_{i=p_3k+1}^{k}(\lambda_i+1-\theta_i)d_i+\sum_{j=p_4k+1}^{p_3k}r_{j,p_2k}(\theta_j-1-\sum_{i=j+1}^{p_2k}\frac{\theta_i}{l_i-i+1})\\&\quad+\sum_{j=1}^{p_4k}r_{j,p_3k}(\theta_j-1-\sum_{i=j+1}^{p_3k}\frac{\theta_i}{l_i-i+1})\end{aligned}$$

Now, if θ_i's satisfy

$$\begin{array}{l}\sum_{i=j+1}^{p_3k}\frac{\theta_i}{l_i-i+1}\ge\theta_j-1\ \text{ if } j\le p_4k\\ \sum_{i=j+1}^{p_2k}\frac{\theta_i}{l_i-i+1}\ge\theta_j-1\ \text{ if } p_4k<j\le p_3k\end{array}\tag{18}$$

This implies

$$\sum_{i=1}^{k}\alpha_i\le\zeta f+\sum_{i=1}^{p_3k}(\lambda_i+2(\theta_i-1))d_i+\sum_{i=p_3k+1}^{k}(\lambda_i+1-\theta_i)d_i\tag{19}$$

Now, if we set the coefficients of d_i's in the right-hand side of the above inequality to γ, we will get the following recurrence for θ_i:

$$\begin{array}{ll}\lambda_i+2(\theta_i-1)=\gamma & \text{if } i\le p_3k\\ \lambda_i+1-\theta_i=\gamma & \text{if } i>p_3k\end{array}\tag{20}$$

Notice that λ_j's can be written as a function of θ_i's ($i\le j$) from Equation 10, thus the above equations are recurrence relations for θ_i.

Solving this recurrence, we get

$$\theta_i=\begin{cases}\frac{\gamma+2}{2}\sqrt{1-\frac{j}{p_2k}} & \text{if } i\le p_1k\\ \theta_{p_1k}\sqrt{1-\frac{j-p_1k}{k-p_1k}} & \text{if } p_1k<i\le p_3k\\ (3-2\theta_{p_3k})\frac{k-p_3k}{k-j} & \text{if } p_3k<i\le p_2k\\ 0 & \text{if } p_2k<i\le k\end{cases}\tag{21}$$

If we can set the constants p_1,p_2,p_3,p_4 in such a way that θ_i's satisfy the Conditions 12, 13, 16, and 18 then Inequality 19 shows that the solution of the factor-revealing LP is at most $\max(\zeta,\gamma)$.

It's not hard to see that θ_i's are decreasing from 0 to p_3k and increasing from p_3k+1 to p_2k, thus in order to satisfy Conditions 13, it is sufficient to have:

$$\theta_{p_3k}\ge1\qquad\text{and}\qquad\theta_{p_2k}\le1\tag{22}$$

In order to write Conditions 12 and 16, we compute the sum of θ_i's for different intervals separately.

$$\sum_{j=1}^{p_3k} \lambda_j = (\gamma+2)p_3k - 2\sum_{j=1}^{p_3k}\theta_j \quad \text{and}$$

$$\begin{aligned}
\sum_{j=1}^{p_3k} \lambda_j &= \sum_{j=1}^{p_3k}\sum_{i=1}^{j} \frac{\theta_i}{l_i-i+1} = \sum_{i=1}^{p_3k}\sum_{j=i}^{p_3k} \frac{\theta_i}{l_i-i+1} = \sum_{i=1}^{p_3k} \frac{(p_3k-i+1)\theta_i}{l_i-i+1} \\
&= \sum_{i=1}^{p_3k}\theta_i + \sum_{i=1}^{p_1k}\frac{(p_3-p_2)k\theta_i}{l_i-i+1} + \sum_{i=p_1k+1}^{p_3k}\frac{(p_3-1)k\theta_i}{l_i-i+1} \\
&= \sum_{i=1}^{p_3k}\theta_i + (p_3-p_2)k\lambda_{p_1k} + (p_3-1)k(\lambda_{p_3k}-\lambda_{p_1k})
\end{aligned}$$

$$\Rightarrow \sum_{i=1}^{p_3k}\theta_i = \frac{k}{3}((\gamma+2)p_3 + (p_2-1)\lambda_{p_1k} + (1-p_3)\lambda_{p_3k})$$

and also

$$\begin{aligned}
\sum_{i=p_3k+1}^{p_2k}\theta_i &= (3-2\theta_{p_3k})(1-p_3)k\sum_{i=(1-p_2)k}^{(1-p_3)k-1}\frac{1}{i} \\
&= (3-2\theta_{p_3k})(1-p_3)k(\ln\frac{1-p_3}{1-p_2} + o(1))
\end{aligned}$$

From these two equations, we can write equation 12 as follows:

$$1 - \frac{\gamma+2}{3}p_3 - \frac{p_2}{3}\lambda_{p_1k} - \frac{1-p_2}{3}\lambda_{p_3k} - (3-2\theta_{p_3k})(1-p_3)\ln\frac{1-p_3}{1-p_2} < 0 \tag{23}$$

In order to write 16 we need the following:

$$\sum_{j=1}^{p_4k} \lambda_j = (\gamma+2)p_4k - 2\sum_{j=1}^{p_4k}\theta_j \quad \text{and}$$

$$\begin{aligned}
\sum_{j=1}^{p_4k} \lambda_j &= \sum_{j=1}^{p_4k}\sum_{i=1}^{j} \frac{\theta_i}{l_i-i+1} = \sum_{i=1}^{p_4k}\sum_{j=i}^{p_4k} \frac{\theta_i}{l_i-i+1} \\
&= \sum_{i=1}^{p_4k} \frac{(p_4k-i+1)\theta_i}{l_i-i+1} = \sum_{i=1}^{p_4k}\theta_i + \sum_{i=1}^{p_4k}\frac{(p_4-p_2)k\theta_i}{l_i-i+1}
\end{aligned}$$

$$\Rightarrow \sum_{i=1}^{p_4k}\theta_i = \frac{k}{3}((\gamma+2)p_4 + (p_2-p_4)\lambda_{p_4k})$$

Therefore, Condition 16 can be written as

$$(p_2-p_3) - (3-2\theta_{p_3k})(1-p_3)\ln\frac{1-p_3}{1-p_2} + o(1) \le \frac{\gamma+2}{3}p_4 + \frac{p_2-p_4}{3}\lambda_{p_4k} - p_4 \tag{24}$$

For $j \le p_4k$, Condition 18 can be written in terms of p_i's and γ as follows:

$$\sum_{i=1}^{p_3k}\frac{\theta_i}{l_i-i+1} - \sum_{i=1}^{j}\frac{\theta_i}{l_i-i+1} \ge \theta_j - 1 \Rightarrow \lambda_{p_3k} - \lambda_j \ge \theta_j - 1 \Rightarrow$$

$$\theta_j \geq 2\theta_{p_3 k} - 1 \qquad \forall j \leq p_4 k$$

In the last inequality, we substituted λ_j in terms of θ_j's using Equation 20. Thus, it is sufficient to have:

$$\theta_{p_4 k} \geq 2\theta_{p_3 k} - 1 \tag{25}$$

Furthermore, for $p_4 k < j \leq p_3 k$

$$\sum_{i=1}^{p_2 k} \frac{\theta_i}{l_i - i + 1} - \sum_{i=1}^{j} \frac{\theta_i}{l_i - i + 1} \geq \theta_j - 1 \Rightarrow \lambda_{p_2 k} - \lambda_j \geq \theta_j - 1 \Rightarrow$$

$$\theta_j + \theta_{p_2 k} \geq 2 \qquad \forall p_4 k < j \leq p_3 k.$$

Again, in the last step we used the Equation 20. Thus, it is sufficient to have:

$$\theta_{p_3 k} + \theta_{p_2 k} \geq 2 \tag{26}$$

The last observation here is that

$$\theta_{p_2 k} \leq 1 \Rightarrow \zeta = \sum_{i=1}^{k} \frac{\theta_i}{l_i - i + 1} = \lambda_{p_2 k} = \gamma - 1 + \theta_{p_2 k} \leq \gamma$$

Thus, from Inequality 22, we get $\zeta \leq \gamma$ and it is sufficient to minimize γ instead of $\max(\zeta, \gamma)$. Now, we need to find $p_4 \leq p_1 \leq p_3 \leq p_2$ such that Inequalities 22, 23, 24, 25 and 26 are all satisfied and γ is minimum. Notice that all these recent inequalities are in terms of p_i's and γ (because θ_i's have been written in terms of them as well as λ_i's). Now we can observe that by setting $p_1 = 0.327$, $p_2 = 0.737$, $p_3 = 0.539$, and $p_4 = 0.327$, all inequalities are satisfied and $\gamma < 1.939 < 1.94$ as desired.