A new greedy approach for facility location problems

Kamal Jain* Mohammad Mahdian† Amin Saberi‡

Abstract

We present a simple and natural greedy algorithm for the metric uncapacitated facility location problem achieving an approximation guarantee of 1.61 whereas the best previously known was 1.73. Furthermore, we will show that our algorithm has a property which allows us to apply the technique of Lagrangian relaxation. Using this property, we can find better approximation algorithms for many variants of the facility location problem, such as the capacitated facility location problem with soft capacities and a common generalization of the k-median and facility location problem. We will also prove a lower bound on the approximability of the k-median problem.

1 Introduction

In the (uncapacitated) facility location problem, we have a set $\mathcal{F}$ of n_f *facilities* and a set $\mathcal{C}$ of n_c *cities*. For every facility $i \in \mathcal{F}$, a nonnegative number f_i is given as the *opening cost* of facility i. Furthermore, for every city $j \in \mathcal{C}$ and facility $i \in \mathcal{F}$, we have a *connection cost* (a.k.a. service cost) c_{ij} between city j and facility i. The objective is to open a subset of the facilities in $\mathcal{F}$, and connect each city to an open facility so that the total cost is minimized. We will consider the *metric* version of this problem, i.e., the connection costs satisfy the triangle inequality.

This problem has many applications in operations research [9, 20], and recently in the network design problems such as placement of routers and caches [14, 21], agglomeration of traffic or data [1, 15], and web server replications in a content distribution network (CDN) [18, 24]. In the last decade the problem was studied extensively from the perspective of approximation algorithms [2, 4, 6, 7, 8, 13, 17, 19, 26, 27].

Different approaches such as LP rounding, primal-dual method, local search, and combinations of these methods with cost scaling and greedy postprocessing are used to solve the facility location problem and its variants. The best known approximation algorithm for this problem, due to Charikar and Guha [4], achieves a factor of 1.728. This algorithm combines the ideas of cost scaling, greedy augmentations, and a primal-dual algorithm of Jain and Vazirani [17] to marginally improve a $(1 + 2/e)$-approximation algorithm of Chudak and Shmoys [7, 8] which is based on LP-rounding techniques. The drawback of this algorithm is that it needs to solve large linear programs and therefore has a high running time. Using the same ideas, Charikar and Guha [4] present an $O(n^3)$ algorithm with approximation ratio 1.853. In [22], it is shown that a classical greedy set-cover algorithm (due to Hochbaum [16]) achieves an approximation ratio of 1.861 in $O(n^2 \log n)$ time. For the case of sparse graph, Thorup [27] gives a faster $(3 + o(1))$-approximation algorithm. Regarding hardness results, Guha and Khuller [13] proved that it is impossible to get an approximation guarantee of 1.463 for the metric facility location problem, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log \log n)}]$. See [25] for a recent survey of the problem.

In this paper, we present a simple and natural heuristic algorithm for the facility location problem, achieving an approximation factor of 1.61 with the running time $O(n^3)$. This algorithm is an improvement of the greedy

*Address: One Microsoft Way, Redmond, WA 98052, USA. E-mail: kamalj@microsoft.com.
†Address: Department of Mathematics, MIT, Cambridge, MA 02139, USA. E-mail: mahdian@math.mit.edu.
‡Address: College of Computing, Georgia Tech, Atlanta, GA 30332, USA. E-mail: saberi@cc.gatech.edu.

algorithm of [22]. The technique used for the analysis of this algorithm is to express different constraints that are imposed by the problem statement or by our algorithm as linear inequalities, so that we can get a bound on the approximation ratio (or in our case, the exact approximation ratio) of the algorithm by solving a particular series of linear programs, which we call factor LP. This scheme is similar to the idea of the LP bound in coding theory (LP bound gives the best known bounds on the minimum distance of a code with a given rate by bounding the solution of a linear program that contains various linear constraints, mainly MacWilliams identities. cf. [23]). It has also been used in the context of approximation algorithms, e.g., by Goemans and Kleinberg [11] for computing the approximation algorithm of their algorithm for the minimum latency problem. This technique enables us to guess the approximation ratio of the algorithm empirically, and provides a straightforward way to prove a bound on the approximation ratio. In the case of our algorithm, this technique also enables us to compute the tradeoff between the approximation ratio of facility costs versus the approximation ratio of the connection costs. The algorithm, its analysis, and a discussion about this tradeoff are presented in Sections 2, 3, and 4, respectively.

Among all known facility location algorithms, the primal-dual algorithm of Jain and Vazirani [17] is perhaps the most versatile one in that it can be used to obtain algorithms for other variants of the problem, such as k-median [17], a common generalization of k-median and facility location [17], capacitated facility location with soft capacities [17], prize collecting facility location [6], and facility location with outliers [6]. This versatility is partly because of a property of the algorithm which allows us to apply the Lagrangian relaxation technique. We will prove in Section 5 that our algorithm has this property (which we call *the Lagrangian multiplier preserving property*) with an approximation factor that beats the primal-dual algorithm. This enables us to obtain algorithms for some variants of the facility location problem. In particular, in this paper we will present an algorithm that solves a common generalization of the facility location problem and k-median within a factor of 4. In this problem, which we refer to as the *k-facility location problem*, an instance of the facility location problem and an integer k are given and the objective is to find the cheapest solution that opens at most k facilities. The k-median problem is a special case of this problem in which all opening costs are 0. The k-median problem is studied extensively [2, 4, 5, 17] and the best known approximation algorithm for this problem, due to Arya et al. [2], achieves a factor of $3 + \epsilon$. The k-facility location problem has also been studied in operations research [9], and the best previously known approximation factor for this problem was 6 [17].

The Lagrangian multiplier preserving property of our algorithm also enables us to give a 3-approximation algorithm for a capacitated version of the facility location problem, in which we are allowed to open more than one facility at any location. We will refer to this problem as the *capacitated facility location problem with soft capacities*. The best previously known approximation algorithm for this problem has a factor of 3.46, and is based on the facility location algorithm of Charikar and Guha [4] together with the observation that any α-approximation algorithm for the uncapacitated facility location problem yields an algorithm with an approximation ratio of 2α for the capacitated facility location problem with soft capacities.

In Section 6, we will prove some lower bounds. We will prove that the k-median problem cannot be approximated within a factor strictly less than $1 + 2/e$, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log \log n)}]$. This is an improvement over a lower bound of $1 + 1/e$ due to Guha [12]. We will also see a lower bound on the best tradeoff we can hope to achieve between the approximation factors for the facility cost and the connection cost in the facility location problem.

2 Algorithm for the metric facility location problem

The facility location problem can be captured by a commonly known integer program due to Balinski [3]. For the sake of convenience, we give another equivalent formulation for the problem. Let us say that a *star* consists of one facility and several cities. The cost of a star is the sum of the opening cost of the facility and the connection costs between the facility and all the cities in the star. Let $\mathcal{S}$ be the set of all stars. The facility location problem can be thought of as picking a minimum cost set of stars such that each city is in at least one star. This problem can be captured by the following integer program. In this program, x_S is an indicator variable denoting whether star S is picked and c_S denotes the cost of star S.

$$\begin{aligned} \text{minimize} \quad & \sum_{S\in\mathcal{S}} c_S x_S \\ \text{subject to} \quad & \forall j \in \mathcal{C}: \sum_{S:j\in S} x_S \geq 1 \\ & \forall S \in \mathcal{S}: \ x_S \in \{0,1\} \end{aligned} \tag{1}$$

The LP-relaxation of this program is:

$$\begin{aligned} \text{minimize} \quad & \sum_{S\in\mathcal{S}} c_S x_S \\ \text{subject to} \quad & \forall j \in \mathcal{C}: \sum_{S:j\in S} x_S \geq 1 \\ & \forall S \in \mathcal{S}: \ x_S \geq 0 \end{aligned} \tag{2}$$

The dual program is:

$$\begin{aligned} \text{maximize} \quad & \sum_{j\in\mathcal{C}} \alpha_j \\ \text{subject to} \quad & \forall S \in \mathcal{S}: \sum_{j\in S\cap\mathcal{C}} \alpha_j \leq c_S \\ & \forall j \in \mathcal{C}: \ \alpha_j \geq 0 \end{aligned} \tag{3}$$

We can think of the variable α_j in the dual program as the share of city j of the total expenses. It is clear from LP-duality that if an algorithm finds a solution for the facility location problem of cost T, and values α_j for $j \in \mathcal{C}$ such that $\sum_{j\in\mathcal{C}} \alpha_j = T$ and for every star S, $\sum_{j\in S\cap\mathcal{C}} \alpha_j \leq \gamma c_S$ for some fixed number $\gamma \geq 1$, then the approximation ratio of the algorithm is at most γ. Another way of looking at this is to consider an optimal solution for an instance of the problem. For every facility i that is opened in this solution and the collection A of cities that are connected to it, we can write the inequality $\sum_{j\in A} \alpha_j \leq \gamma(f_i + \sum_{j\in A} c_{ij})$. By adding up these inequalities, we will find out that the cost of our solution is at most γ times the cost of the optimal solution. This fact is the basis of our analysis.

Algorithm 1

1. We introduce a notion of time, as in [17]. The algorithm starts at time 0. At this time, all cities are unconnected, all facilities are closed, and the *budget* of every city j, denoted by B_j, is initialized to 0. At every moment, each city j offers some money from its budget to each *closed* facility i. The amount of this offer is computed as follows: If j is unconnected, the offer is equal to $\max(B_j - c_{ij}, 0)$ (i.e., if the budget of j is more than the cost that it has to pay to get connected to i, it offers to pay this extra budget to i); If j is already connected to some other facility i', then its offer to facility i is equal to $\max(c_{i'j} - c_{ij}, 0)$ (i.e., the amount that j offers to pay to i is equal to the money that it would save if it switches its facility from i' to i).

2. While there is an unconnected city, increase the time, and simultaneously, increase the budget of each *unconnected* city at the same rate (i.e., every unconnected city j has $B_j = t$ at time t), until one of the following events occur. If multiple events occur at the same time, process them in an arbitrary order.

(a) For some closed facility i, the total offer that it receives from cities is equal to the cost of opening i. In this case, we open facility i, and for every city j (connected or unconnected) which has a non-zero offer to i, we connect j to i. The amount that j had offered to i is now called the *contribution* of j toward i, and j is no longer allowed to decrease this contribution.

(b) For some unconnected city j, and some facility i that is already open, the budget of j is equal to the connection cost between j and i. In this case, we connect city j to facility i. The contribution of j toward i is zero.

Notice that once a city gets connected, we stop increasing its budget. Also, it should be obvious that the budget of each connected city is always equal to the connection cost that it pays at the time, plus the total contribution that it has given to the facilities.

3. For every city j, set α_j (the share of j of the total expenses) equal to the budget of j at the end of algorithm. Notice that this value is also equal to the time that j first gets connected.

The following fact should be obvious from the description of the algorithm.

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

3 Analysis of the algorithm

In this section we compute the approximation ratio of Algorithm 1. By the comments before Algorithm 1, we know that in order to prove an approximation guarantee of γ, it is enough to show that for every star S, the sum of α_j's of the cities in S is at most γ times the cost of S. In order to compute such a γ, in Section 3.1 we will define an optimization program (called the factor LP) whose solution gives the value of γ. In Section 3.2 we will use the factor LP to prove an upper bound of 1.61 on the approximation ratio of Algorithm 1. In Section 3.3 we show that our analysis is tight.

3.1 Deriving the factor LP

In this section, we express various constraints that are imposed by the problem or by the structure of the algorithm as inequalities, so that we can get a bound on the value of γ defined above by solving a series of linear programs.

Consider a star S consisting of a facility of opening cost f (with a slight misuse of the notation, we call this facility f), and k cities numbered 1 through k. Let d_j denote the connection cost between facility f and city j, and α_j denote the share of j of the expenses, as defined in Algorithm 1. We may assume without loss of generality that

$$\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_k. \tag{4}$$

We need more variables to capture the execution of Algorithm 1. For every i $(1 \leq i \leq k)$, consider the situation of the algorithm at time $t = \alpha_i - \epsilon$, where ϵ is very small, i.e., just a moment before city i gets connected for the first time. At this time, each of the cities $1, 2, \ldots, i-1$ might be connected to a facility. For every $j < i$, if city j is connected to some facility at time t, let $r_{j,i}$ denote the connection cost between this facility and city j; otherwise, let $r_{j,i} := \alpha_j$. Obviously, the latter case occurs if and only if $\alpha_i = \alpha_j$. It turns out that these variables (f, d_j's, α_j's, and $r_{j,i}$'s) are enough to write down some inequalities to bound the ratio of the sum of α_j's to the cost of S (i.e., $f + \sum_{j=1}^{k} d_j$).

First, notice that once a city gets connected to a facility, its budget remains the same and it cannot take back its contribution to a facility, so it can never get connected to another facility with a higher connection cost. This implies that for every j,

$$r_{j,j+1} \geq r_{j,j+2} \geq \cdots \geq r_{j,k}. \tag{5}$$

Now, consider the time $t = \alpha_i - \epsilon$. At this time, the amount of offer of city j toward facility f is equal to

$$\begin{array}{ll} \max(r_{j,i} - d_j, 0) & \text{if } j < i, \text{ and} \\ \max(t - d_j, 0) & \text{if } j \geq i. \end{array}$$

(Notice that this holds even if $j < i$ and $\alpha_i = \alpha_j$.) It is clear from Algorithm 1 that the total offer of cities to a facility can never become larger than the opening cost of the facility. Therefore, we have the following inequality.

$$\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f. \tag{6}$$

Another important constraint that we need to use is the triangle inequality. By the triangle inequality and the definition of $r_{j,i}$, we know that for every $j < i$, the connection cost between city i and the facility to which city j is connected at time $t = \alpha_i - \epsilon$ (let's call this facility f') is at most $r_{j,i} + d_i + d_j$. This cost cannot be less than t, since if it is, our algorithm could have connected the city i to the facility f' at a time earlier than t, which is a contradiction. (Here we need to be careful with the special case $\alpha_i = \alpha_j$. In this case, $r_{j,i} + d_i + d_j$ is obviously not more than t. If $\alpha_i \neq \alpha_j$, the facility f' is open at time t and therefore city i can get connected to it, if it can pay the connection cost). This argument shows that for every $1 \leq j < i \leq k$,

$$\alpha_i \leq r_{j,i} + d_i + d_j. \tag{7}$$

The above inequalities form the following optimization program, which we call the factor LP. Notice that although the following optimization program is not written in the form of a linear program, it is easy to change it to a linear program by introducing new variables and inequalities.

$$\begin{array}{lll} z_k = & \text{maximize} & \dfrac{\sum_{i=1}^{k} \alpha_i}{f + \sum_{i=1}^{k} d_i} \\ & \text{subject to} & \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_k \\ & & r_{j,j+1} \geq r_{j,j+2} \geq \cdots \geq r_{j,k} \qquad \forall\, 1 \leq j \leq k \\ & & \alpha_i \leq r_{j,i} + d_i + d_j \qquad \forall\, 1 \leq j < i \leq k \\ & & \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f \qquad \forall\, 1 \leq i \leq k \\ & & \alpha_j, d_j, f, r_{j,i} \geq 0 \qquad \forall\, 1 \leq j \leq i \leq k \end{array} \tag{8}$$

Now, we can state the following lemma.

Lemma 2 *If z_k denotes the solution of the factor LP, then for every star S consisting of a facility and k cities, the sum of α_j's of the cities in S in Algorithm 1 is at most $z_k c_S$.*

Proof : Inequalities 4, 5, 6, and 7 derived above imply that the values $\alpha_j, d_j, f, r_{j,i}$ that we get by running Algorithm 1 constitute a feasible solution of the factor LP. Thus, the value of the objective function for this solution is at most z_k. □

Lemmas 1 and 2 imply the following.

k	$\max_{i \le k} z_i$
10	1.54147
20	1.57084
50	1.58839
100	1.59425
200	1.59721
300	1.59819
400	1.59868
500	1.59898

Table 1: Solution of the factor LP

Lemma 3 *Let z_k be the solution of the factor LP, and $\gamma := \sup_k\{z_k\}$. Then Algorithm 1 is a γ-approximation algorithm for the facility location problem.*

3.2 Solving the factor LP

As mentioned earlier, the optimization program 8 can be written as a linear program. This enables us to use an LP-solver to solve the factor LP for small values of k, in order to guess the numerical value of γ. Table 1 shows a summary of results that are obtained by solving the factor LP using the software AMPL. It seems from the experimental results that z_k is an increasing sequence that converges to some number close to 1.6 and hence $\gamma \approx 1.6$.

By solving the factor LP for any particular value of k, we get a lower bound on the value of γ. In order to prove an upper bound on γ, we need to present a general solution to the dual of the factor LP. Unfortunately, this is not an easy task in general. (For example, performing a tight asymptotic analysis of the LP bound is still an open question in coding theory). However, here empirical results can help us: we can solve the dual of the factor LP for small values of k to get an idea how the general optimal solution looks like. Using this, it is usually possible (although sometimes tedious) to prove a close-to-optimal upper bound on the value of z_k.

Here we have used this technique to prove an upper bound of 1.61 on the value of γ. The proof is presented in Appendix A.

3.3 Tightness result

In this section, we show our analysis of the approximation ratio of Algorithm 1 in the previous sections is tight. Numerical results shown in Table 1 show that the upper bound 1.61 on the solution z_k of the factor LP is very close to the best upper bound. The only thing that remains is to show that we can actually construct an example on which our algorithm performs at least z_k times worse than the optimum. The construction is based on the following observation, which is motivated by the experimental results. We omit the proof of this fact here, as it is long and technical.

Lemma 4 *There is an optimal solution for the factor LP in which for every $j < i$, $r_{j,i} = d_j \le \alpha_j$.*

Now we can prove the following lemma. The proof is presented in Appendix B

Lemma 5 *For every k, there is an instance of the facility location problem with k cities and $k+1$ facilities for which Algorithm 1 outputs a solution of cost at least z_k times the optimum solution.*

Therefore, the above results and an easy observation about the running time of Algorithm 1 imply the following.

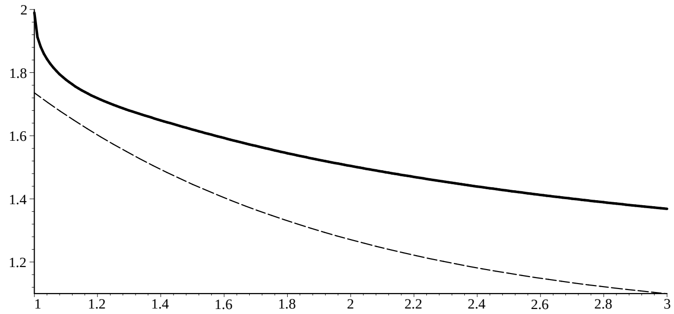

Figure 1: The tradeoff between γ_f and γ_c

Theorem 6 *Algorithm 1 solves the facility location problem in time $O(n^3)$, where $n = \max(n_f, n_c)$. Its approximation ratio is* equal *to the supremum of the solution of the maximization program 8, which is less than 1.61, and more than 1.598.*

4 The tradeoff between facility and connection costs

We defined the cost of a solution in the facility location problem as the sum of the facility cost (i.e., total cost of opening facilities) and the connection cost. We proved in the previous section that Algorithm 1 achieves an overall performance guarantee of 1.61. However, sometimes it is useful to get different approximation guarantees for facility and connection costs. The following theorem gives such a guarantee. The proof is similar to the proof of Lemma 3.

Theorem 7 *Let $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program.*

$$
\begin{array}{lll}
z_k = & \text{maximize} & \dfrac{\sum_{i=1}^k \alpha_i - \gamma_f f}{\sum_{i=1}^k d_i} \\
& \text{subject to} & \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_k \\
& & r_{j,j+1} \geq r_{j,j+2} \geq \cdots \geq r_{j,k} \qquad \forall 1 \leq j \leq k \\
& & \alpha_i \leq r_{j,i} + d_i + d_j \qquad \forall 1 \leq j < i \leq k \\
& & \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f \qquad \forall 1 \leq i \leq k \\
& & \alpha_j, d_j, f, r_{j,i} \geq 0 \qquad \forall 1 \leq j \leq i \leq k
\end{array}
\tag{9}
$$

Then for every instance $\mathcal{I}$ of the facility location problem, and for every solution SOL for $\mathcal{I}$ with facility cost F_{SOL} and connection cost C_{SOL}, the cost of the solution found by Algorithm 1 is at most $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.

We have computed the solution of the optimization program 9 for $k = 100$, and several values of γ_f between 1 and 3, to get an estimate of the corresponding γ_c's. The result is shown in the diagram in Figure 1. Every point (γ_f, γ_c') on the thick line in this diagram represents a value of γ_f, and the corresponding estimate for the value of γ_c. The dashed line shows a lower bound that holds unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$ and is proved in Section 6

Similar tradeoff problems are considered in [4]. However, an important advantage that we get here is that all the inequalities $ALG \leq \gamma_f F_{SOL} + \gamma_c C_{SOL}$ are satisfied by a *single* algorithm.

We will see in the next section that the case $\gamma_f = 1$ is of particular theoretical importance for designing other algorithms.

5 The metric k-facility location problem

The k-median problem differs from the facility location problem in two respects: there is no cost for opening facilities, and there is an upper bound k, that is supplied as part of the input, on the number of facilities that can be opened. Jain and Vazirani [17] reduced this problem to the facility location problem in the following sense: Suppose $\mathcal{A}$ is an approximation algorithm for the facility location problem. Consider an instance $\mathcal{I}$ of the problem with optimum cost OPT, and let F and C be the facility and connection costs of the solution found by $\mathcal{A}$. Algorithm $\mathcal{A}$ is called a Lagrangian Multiplier Preserving α-approximation (or LMP α-approximation for short) if for every instance $\mathcal{I}$, $C \leq \alpha(OPT - F)$. Jain and Vazirani [17] show that an LMP α-approximation algorithm for the metric facility location problem gives rise to a 2α-approximation algorithm for the metric k-median problem. Their theorem also holds for a common generalization of the metric k-median and the metric facility location problem. In this problem one has the cost on the facilities as well as the upper bound of k on the number of facilities which can be opened. We will refer this problem as the *metric k-facility location problem.* We give here an LMP 2-approximation algorithm for the metric facility location problem. This will result in a 4-approximation algorithm for the metric k-facility location problem, whereas the best previously known was 6 [17].

Lemma 8 **[17]** *An LMP α-approximation algorithm for the facility location problem gives a 2α-approximation algorithm for the k-facility location problem.*

Jain and Vazirani show the versatility of LMP approximation algorithms by giving a 4-approximation algorithm for the metric capacitated facility location problem in which opening multiple copies of the same facility is allowed. Formally, an instance of capacitated facility location problem is similar to an instance of uncapacitated problem except that for every facility i, there is one more parameter, u_i, which indicates the *capacity* of this facility, i.e., the number of cities it can serve. We will refer to the version of this problem in which we are allowed to open each facility more than once as the *capacitated facility location problem with soft capacities.* In general, this problem is reducible to the uncapacitated facility location problem within a factor of 2; change the connection cost between a facility i and a city j from c_{ij} to $c_{ij} + \frac{f_i}{u_i}$. This reduction together with the algorithm of Charikar and Guha [4] yields a 3.46-approximation algorithm for the metric capacitated problem, which was the best previously known algorithm. However, one can easily generalize the Jain-Vazirani [17] section on capacitated facility location problem to the following lemma. This lemma, together with our LMP 2-approximation algorithm for the facility location problem gives a 3-approximation algorithm for the capacitated facility location problem with soft capacities.

Lemma 9 *An LMP α-approximation algorithm for the metric uncapacitated facility location problem leads to an $(\alpha+1)$-approximation algorithm for the metric capacitated facility location problem with soft capacities.*

In this section, we show that there is an LMP 2-approximation algorithm for the metric facility location problem. The proof is based on Theorem 7 together with the scaling technique of [4]. We prove the following lemma using this technique.

Lemma 10 *Assume there is an algorithm $\mathcal{A}$ for the metric facility location problem that for every instance $\mathcal{I}$ and every solution SOL for $\mathcal{I}$, $\mathcal{A}$ finds a solution of cost at most $F_{SOL} + \alpha C_{SOL}$, where F_{SOL} and C_{SOL} are facility and connection costs of SOL, and α is a fixed number. Then there is an LMP α-approximation algorithm for the metric facility location problem.*

Proof : Consider the following algorithm: The algorithm constructs another instance $\mathcal{I}'$ of the problem by multiplying the facility opening costs by α, runs Algorithm 1 on this modified instance $\mathcal{I}'$, and outputs its answer. Suppose αF (F with the original costs) and C be the facility and the connection costs in the solution provided by this run. Then $\alpha F + C \leq \alpha(F_{SOL} + C_{SOL})$, which implies that this algorithm is an LMP α-approximation. $\square$

Now we only need to prove the following. The proof of this theorem is presented in Appendix C.

Theorem 11 *For every instance $\mathcal{I}$ and every solution SOL for $\mathcal{I}$, Algorithm 1 finds a solution of cost at most $F_{SOL} + 2C_{SOL}$, where F_{SOL} and C_{SOL} are facility and connection costs of SOL.*

Lemma 10 and Theorem 11 provide an LMP 2-approximation algorithm for the metric facility location problem. This result improves all the results in [17], and gives straightforward algorithms for some other problems considered by Charikar et al [6].

6 Lower bounds

In this section we explore some impossibility results. We first prove that metric k-median can not be approximated within a factor strictly smaller than $1 + \frac{2}{e}$, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. This improves a lower bound of $1 + \frac{1}{e}$ due to Guha [12]. Our method is similar to the technique used by Guha and Khuller [13] to prove the hardness of the metric facility location problem. We will also adapt their proof [13] to show a lower bound on the tradeoff results. This gives a lower bound on the best approximation factor that one can hope to achieve for the k-median problem using the standard Lagrangian relaxation method. Proofs are presented in Appendix D.

Theorem 12 *Metric k-median problem cannot be approximated within a factor strictly smaller than $1 + \frac{2}{e}$ unless minimum set-cover can be approximated within a factor of $c \ln n$ for $c < 1$.*

The above theorem together with Feige's result on the hardness of set-cover [10] establishes the hardness of $(1+\frac{2}{e}-\epsilon)$-approximation for k-median. This proves that k-median is a strictly harder problem to approximate than the facility location problem because the latter can be approximated within a factor of 1.61.

Theorem 13 *Let γ_f and γ_c be constants with $\gamma_c < 1 + 2e^{-\gamma_f}$. Assume there is an algorithm $\mathcal{A}$ that for every instance $\mathcal{I}$ of the metric facility location problem finds a solution whose cost is not more than $\gamma_f F_{SOL} + \gamma_c C_{SOL}$ for every solution SOL for $\mathcal{I}$ with facility and connection costs F_{SOL} and C_{SOL}. Then minimum set-cover can be approximated within a factor of $c \ln n$ for $c < 1$.*

The dashed line in Figure 1 shows the lower bound provided by the above theorem.

The above theorem shows that finding an LMP $(1 + \frac{2}{e} - \epsilon)$-approximation for the metric facility location problem is hard. Also, the integrality gap examples presented in [12] show that Lemma 8 from [17] is tight. This shows that one cannot use Lemma 8 as a black box to obtain a smaller factor than $2 + \frac{4}{e}$ for k-median problem. Note that $3 + \epsilon$ approximation is already known [2] for the problem. Hence if one wants to beat this factor using the Lagrangian relaxation technique then it will be necessary to look into the underlying LMP algorithm as already been done in [4].

7 Concluding Remarks

The method that we used in this paper seems to be a general tool for analyzing greedy and heuristic algorithms. It would be interesting to find other examples that can be analyzed using this method.

When analyzing a problem using this method, we usually encounter the problem of finding the limit of the solution of a sequence of linear programs. This seems to be a very difficult task in general. It would be nice if one can develop a general method for solving such problems. One possible idea is to consider the limit of the values of the variables in the optimal solution as a continuous function, and derive functional equations from the inequalities.

We have implemented our facility location algorithm, and run it on several randomly generated test cases. The test cases were generated by considering shortest distance metric in a complete bipartite graph with random weights, or Euclidean distance metric among a set of random points as the connection costs. In all instances, the solution given by our algorithm was at most a factor of 1.05 away from the lower bound obtained by solving the LP relaxation of the problem. This shows that in practice our algorithm works much better than the guaranteed approximation ratio. A theoretical explanation of this fact would be interesting.

Acknowledgment. We would like to thank Vijay Vazirani, Evangelos Markakis, Nicole Immorlica, Milena Mihail, and Michel Goemans for their helpful comments.

References

[1] M. Andrews and L. Zhang. The access network design problem. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, 2001.

[3] M.L. Balinski. On finding integer solutions to linear programs. In *Proc. IBM Scientific Computing Symposium on Combinatorial Problems*, pages 225–248, 1966.

[4] 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.

[5] M. Charikar, S. Guha, E. Tardos, and D.B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 1–10, May 1999.

[6] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. Facility location with outliers. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, Washington DC, January 2001.

[7] 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.

[8] F.A. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. unpublished manuscript, 1998.

[9] G. Cornuejols, G.L. Nemhauser, and L.A. Wolsey. The uncapacitated facility location problem. In P. Mirchandani and R. Francis, editors, *Discrete Location Theory*, pages 119–171. John Wiley and Sons Inc., 1990.

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

[11] M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.

[12] S. Guha. *Approximation algorithms for facility location problems.* PhD thesis, Stanford University, 2000.

[13] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31:228–248, 1999.

[14] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[15] S. Guha, A. Meyerson, and K. Munagala. Improved combinatorial algorithms for single sink edge installation problems. Technical Report STAN-CS-TN00 -96, Stanford University, 2000.

[16] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(2):148–162, 1982.

[17] 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.

[18] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentations. In *Proceedings of IEEE INFOCOM'00*, pages 26–30, March 2000.

[19] 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.

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

[21] B. Li, M. Golin, G. Italiano, Deng X., and K. Sohraby. On the optimal placement of web proxies in the internet. In *Proceedings of IEEE INFOCOM'99*, pages 1282–1290, 1999.

[22] M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. A greedy facility location algorithm analyzed using dual fitting. manuscript.

[23] R.J. McEliece, E.R. Rodemich, H. Rumsey Jr., and L.R. Welch. New upper bounds on the rate of a code via the delsarte-macwilliams inequalities. *IEEE Transactions on Information Theory*, 23:157–166, 1977.

[24] L. Qiu, V.N. Padmanabhan, and G. Voelker. On the placement of web server replicas. In *Proceedings of IEEE INFOCOM'01*, Anchorage, AK, USA, April 2001.

[25] D.B. Shmoys. Approximation algorithms for facility location problems. In K. Jansen and S. Khuller, editors, *Approximation Algorithms for Combinatorial Optimization*, volume 1913 of *Lecture Notes in Computer Science*, pages 27–33. Springer, Berlin, 2000.

[26] 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.

[27] M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. In *Automata, Languages and Programming, 28th International Colloquium, Crete, Greece*, volume 2076 of *Lecture Notes in Computer Science*, pages 249–260, 2001.

A Proof of the upper bound on the solution of the factor LP

We start by proving the following lemma which allows us to concentrate on the case when k is sufficiently large.

Lemma 14 *If z_k denotes the solution to the factor LP, then for every k, $z_k \le z_{2k}$.*

Proof : Let $(\alpha_j, d_j, f, r_{j,i})$ be the optimum solution of the factor LP for k. We construct a feasible solution $(\alpha'_j, d'_j, f', r'_{j,i})$ for $2k$ by duplicating everything as follows: $\alpha'_{2j-1} = \alpha'_{2j} = \alpha_j$, $r'_{2j-1,2i-1} = r'_{2j-1,2i} = r'_{2j,2i-1} = r'_{2j,2i} = r_{j,i}$, $d'_{2j-1} = d'_{2j} = d_j$, and $f' = 2f$. It is easy to see that this is a feasible solution for $2k$ with an objective value of z_k. Thus, $z_{2k} \geq z_k$. $\square$

Lemma 15 *Let z_k be the solution to the factor LP. Then for every sufficiently large k, $z_k \leq 1.61$.*

Proof : Consider a feasible solution of the factor LP. Let $x_{j,i} := max(r_{j,i} - d_j, 0)$. The fourth inequality of the factor LP implies that for every $i \leq i'$,

$$(i' - i + 1)\alpha_i \leq \sum_{j=i}^{i'} d_j + f - \sum_{j=1}^{i-1} x_{j,i}. \tag{10}$$

Now, we define l_i as follows:

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

where p_1 and p_2 are two constants (with $p_1 < p_2$) that will be fixed later. Consider Inequality 10 for every $i \leq p_2 k$ and $i' = l_i$, and divide both sides of this inequality by $(l_i - i + 1)$. By adding up these inequalities we obtain:

$$\sum_{i=1}^{p_2 k} \alpha_i \leq \sum_{i=1}^{p_2 k} \sum_{j=i}^{l_i} \frac{d_j}{l_i - i + 1} + (\sum_{i=1}^{p_2 k} \frac{1}{l_i - i + 1})f - \sum_{i=1}^{p_2 k} \sum_{j=1}^{i-1} \frac{x_{j,i}}{l_i - i + 1} \tag{11}$$

Now for every $j \leq p_2 k$, let $y_j := x_{j,p_2 k}$. The second inequality of the factor LP implies that $x_{j,i} \geq y_j$ for every $j < i \leq p_2 k$ and $x_{j,i} \leq y_j$ for every $i > p_2 k$. Also, let $\zeta := \sum_{i=1}^{p_2 k} \frac{1}{l_i - i + 1}$. Therefore, inequality 11 implies

$$\sum_{i=1}^{p_2 k} \alpha_i \leq \sum_{i=1}^{p_2 k} \sum_{j=i}^{l_i} \frac{d_j}{l_i - i + 1} + \zeta f - \sum_{i=1}^{p_2 k} \sum_{j=1}^{i-1} \frac{y_j}{l_i - i + 1} \tag{12}$$

Consider the index $\ell \leq p_2 k$ for which $2d_\ell + y_\ell$ has its minimum (i.e., for every $j \leq p_2 k$, $2d_\ell + y_\ell \leq 2d_j + y_j$). The third inequality of the factor LP implies that

$$\alpha_i \leq r_{\ell,i} + d_i + d_\ell \leq x_{\ell,i} + 2d_\ell + d_i \leq d_i + 2d_\ell + y_\ell \qquad \text{for } i = p_2 k + 1, \ldots, k \tag{13}$$

By adding Inequality 13 for $i = p_2 k + 1, \ldots, k$ with Inequality 12 we obtain

$$\begin{aligned} \sum_{i=1}^{k} \alpha_i \leq & \sum_{i=1}^{p_2 k} \sum_{j=i}^{l_i} \frac{d_j}{l_i - i + 1} + (2d_\ell + y_\ell)(1 - p_2)k + \sum_{j=p_2 k+1}^{k} d_j - \sum_{i=1}^{p_2 k} \sum_{j=1}^{i-1} \frac{y_j}{l_i - i + 1} + \zeta f \\ = & \sum_{j=1}^{p_2 k} \sum_{i=1}^{j} \frac{d_j}{l_i - i + 1} + \sum_{j=p_2 k+1}^{k} (1 + \sum_{i=p_1 k+1}^{p_2 k} \frac{1}{k - i + 1}) d_j \\ & + (2d_\ell + y_\ell)(1 - p_2)k - \sum_{j=1}^{p_2 k} \sum_{i=j+1}^{p_2 k} \frac{y_j}{l_i - i + 1} + \zeta f \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{p_2k} \zeta d_j - \sum_{j=1}^{p_2k} \sum_{i=j+1}^{p_2k} \frac{d_j + y_j}{l_i - i + 1} + \sum_{j=p_2k+1}^{k} (1 + \sum_{i=p_1k+1}^{p_2k} \frac{1}{k-i+1}) d_j + (2d_\ell + y_\ell)(1-p_2)k + \zeta f \\
&\le \sum_{j=1}^{p_2k} \zeta d_j + (2d_\ell + y_\ell) \left((1-p_2)k - \frac{1}{2} \sum_{j=1}^{p_2k} \sum_{i=j+1}^{p_2k} \frac{1}{l_i - i + 1} \right) \\
&\quad + \sum_{j=p_2k+1}^{k} (1 + \sum_{i=p_1k+1}^{p_2k} \frac{1}{k-i+1}) d_j + \zeta f,
\end{aligned}
$$

where the last inequality is a consequence of the inequality $2d_\ell + y_\ell \le 2d_j + y_j \le 2d_j + 2y_j$ for $j \le p_2k$. Now, let $\zeta' := 1 + \sum_{i=p_1k+1}^{p_2k} \frac{1}{k-i+1}$ and $\delta := (1-p_2) - \frac{1}{2k} \sum_{j=1}^{p_2k} \sum_{i=j+1}^{p_2k} \frac{1}{l_i - i+1}$. Therefore, the above inequality can be written as follows:

$$
\sum_{i=1}^{k} \alpha_i \le \sum_{j=1}^{p_2k} \zeta d_j + \sum_{j=p_2k+1}^{k} \zeta' d_j + \delta(2d_\ell + y_\ell)k + \zeta f, \tag{14}
$$

where

$$
\begin{aligned}
\zeta &= \sum_{i=1}^{p_2k} \frac{1}{l_i - i + 1} \\
&= \sum_{i=1}^{p_1k} \frac{1}{p_2k - i + 1} + \sum_{i=p_1k+1}^{p_2k} \frac{1}{k-i+1} \\
&= H_{p_2k} - H_{p_2k - p_1k} + H_{k-p_1k} - H_{k-p_2k} \\
&= \ln \frac{p_2(1-p_1)}{(p_2-p_1)(1-p_2)} + o(1), && (15) \\
\zeta' &= 1 + \sum_{i=p_1k+1}^{p_2k} \frac{1}{k-i+1} \\
&= 1 + H_{k-p_1k} - H_{k-p_2k} \\
&= 1 + \ln \frac{1-p_1}{1-p_2} + o(1), \text{ and} && (16) \\
\delta &= 1 - p_2 - \frac{1}{2k} \sum_{j=1}^{p_2k} \sum_{i=j+1}^{p_2k} \frac{1}{l_i - i + 1} \\
&= 1 - p_2 - \frac{1}{2k} \sum_{i=1}^{p_2k} \frac{i-1}{l_i - i + 1} \\
&= 1 - p_2 + \frac{1}{2k} \left(p_2k - p_2k(H_{p_2k} - H_{(p_2-p_1)k}) - k(H_{(1-p_1)k} - H_{(1-p_2)k}) \right) \\
&= \frac{1}{2}(2 - p_2 - p_2 \ln \frac{p_2}{p_2 - p_1} - \ln \frac{1-p_1}{1-p_2}) + o(1). && (17)
\end{aligned}
$$

Now if we choose p_1 and p_2 such that $\delta < 0$, and let $\gamma := \max(\zeta, \zeta')$ then inequality 14 implies that

$$
\sum_{i=1}^{k} \alpha_i \le (\gamma + o(1))(f + \sum_{i=1}^{k} d_j).
$$

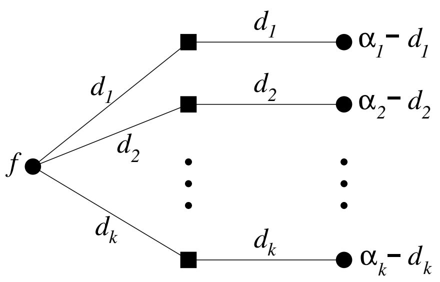

Figure 2: Tightness example. Cities and facilities are marked by squares and circles, respectively.

Using equations 15, 16, and 17, it is easy to see that subject to the condition $\delta < 0$, the value of γ is minimized when $p_1 \approx 0.439$ and $p_2 \approx 0.695$, which gives us $\gamma < 1.61$. □

B Proof of Lemma 5

Proof : By Lemma 4, there is an optimal feasible solution of the factor LP. Consider this solution. We construct an instance of the facility location problem as follows: There are k cities and $k+1$ facilities. The cost of opening facility i is

$$f_i = \begin{cases} \alpha_i - d_i & \text{if } 1 \le i \le k \\ f & \text{if } i = k+1. \end{cases}$$

For every $1 \le j \le k$, the connection cost between city j and facility j and the connection cost between city j and facility $k+1$ are equal to d_j. Other connection costs are implied by the metric inequality, i.e., the connection cost between city j and facility i ($i \ne j, 1 \le i, j \le k$) is equal to $2d_i + d_j$. This example is shown in Figure 2.

On this instance, for every city i, since $c_{ii} + f_i = \alpha_i$, at time α_i Algorithm 1 connects city i to facility i, if it is not already connected to another facility. Thus, the budget of city i cannot exceed α_i. On the other hand, by the third inequality of the factor LP, $\alpha_i \le 2d_j + d_i = c_{j,i}$ for every $i < j$. Therefore, the budget of city i is never enough for connecting to facilities other than facilities i and $k+1$. Also, the fourth inequality of the factor LP guarantees that city i can get connected to facility i before facility $k+1$ can be opened. Therefore, for every i, Algorithm 1 connects city i to facility i, and the cost of this solution is equal to $\sum_{i=1}^{k} c_{ii} + \sum_{i=1}^{k} f_i = \sum_{i=1}^{k} \alpha_i$.

On the other hand, the optimal solution for this instance is to connect all the cities to facility $k+1$. Cost of this solution is equal to $f_{k+1} + \sum_{i=1}^{k} c_{i,k+1} = f + \sum_{i=1}^{k} d_i = z_k \times \sum_{i=1}^{k} \alpha_i$.

Therefore, Algorithm 1 outputs a solution whose cost is at least z_k times the cost of the optimal solution. □

C Proof of Theorem 11

The proof of Theorem 11 is a simple corollary of Theorem 7 and the following Lemma.

Lemma 16 *If z_k is the solution of the maximization program 9 for $\gamma_f = 1$, then $z_k = 2 - \frac{1}{k}$.*

Proof : First, consider the following solution for the program 9.

$$\alpha_i = \begin{cases} 2 - \frac{1}{k} & i = 1 \\ 2 & 2 \le i \le k \end{cases}$$
$$d_i = \begin{cases} 1 & i = 1 \\ 0 & 2 \le i \le k \end{cases}$$
$$r_{j,i} = \begin{cases} 1 & j = 1 \\ 2 & 2 \le j \le k \end{cases}$$
$$f = 2(k-1)$$

It's easy to verify that this is a feasible solution whose objective value is $2 - \frac{1}{k}$.
Now, we prove that every feasible solution of program 9 has an objective value less than or equal to $2 - \frac{1}{k}$. Consider a feasible solution. From the fourth inequality of the factor LP we have:

$$\sum_{j=1}^{i-1}(r_{j,i} - d_j) + \sum_{j=i}^{k}(\alpha_i - d_j) \le f$$

This, together with the third inequality of the factor LP implies that

$$\sum_{j=1}^{i-1}(\alpha_i - d_i - 2d_j) + \sum_{j=i}^{k}(\alpha_i - d_j) \le f \Rightarrow k\alpha_i - f \le (i-1)d_i + \sum_{j=1}^{k} d_j + \sum_{j=1}^{i-1} d_j$$

Adding the above inequality for $i = 1, \ldots, k$, we obtain

$$\begin{aligned} k(\sum_{i=1}^{k}\alpha_i - f) &\le \sum_{i=1}^{k}(i-1)d_i + k\sum_{j=1}^{k} d_j + \sum_{i=1}^{k}\sum_{j=1}^{i-1} d_j \\ &= \sum_{i=1}^{k}(i-1)d_i + k\sum_{j=1}^{k} d_j + \sum_{j=1}^{k}(k-j)d_j \\ &= (2k-1)\sum_{j=1}^{k} d_j \end{aligned}$$

which completes the proof. □

Remark. It is possible to construct an example which shows that the bound in Theorem 11 is tight. The construction is similar to the one in the proof of Lemma 5.

D Proof of lower bounds

Proof of Theorem 12 : Suppose an instance, $(X, \mathcal{S})$, of the set-cover problem is given, where X is the set of elements to be covered and $\mathcal{S}$ is the collection of available sets. Suppose we guess that the minimum number of sets needed to cover X is k (this we can assume because we can try all guesses from 1 to $|X|$). We create an instance of k-median as follows. We choose the k of k-median to be our guess. We create a facility for each set $S \in \mathcal{S}$, and a city for each element $i \in X$. The connection cost between S and i is 1 if $i \in S$ and 3 otherwise. From our guess we know that the optimum cost for this instance of k-median is $|X|$.

Now suppose that k-median can be approximated within a factor α. Take an α factor feasible solution for the instance. We say that a city is served *directly* when the connection cost for it is 1; otherwise it is served *indirectly*. Clearly any α factor feasible solution can serve at most $\frac{\alpha-1}{2}$ fraction of cities indirectly. So we have picked k sets which covered all elements except at most $\frac{\alpha-1}{2}$ fraction. Keep the same k and form another instance of k-median to cover these elements. We will have to repeat this at most $\log_{\frac{2}{\alpha-1}} |X| = (\ln|X|)/\ln(\frac{2}{\alpha-1})$ times. Hence $\ln(\frac{2}{\alpha-1}) < 1$. This gives us $\alpha \geq 1 + \frac{2}{e}$. $\square$

Proof of Theorem 13 : This proof is on the lines of Guha-Khuller proof on the hardness of facility location problem [13]. Consider an instance, $(X, \mathcal{S})$, of the set-cover problem. Guess that the minimum number of sets needed to cover X is k. Create an instance $(\mathcal{S}, X)$ of the facility location problem as done in the proof of Theorem 12 with facility costs $f_S = \zeta\frac{|X|}{k}$ for all S, where $\zeta < 1$ is a constant chosen later. Because there is a set-cover of size k, we can choose $F_{SOL} = \zeta|X|$ and $C_{SOL} = |X|$. Therefore, algorithm $\mathcal{A}$ finds a solution of cost at most $(\zeta\gamma_f + \gamma_c)|X|$. Suppose this solution opens βk facilities and serves $\delta|X|$ cities indirectly. Thus, the cost of this solution is $\beta k\zeta\frac{|X|}{k} + 3\delta|X| + (1-\delta)|X|$. Therefore,

$$\beta\zeta + 2\delta + 1 \leq \zeta\gamma_f + \gamma_c. \tag{18}$$

We claim that there is a constant $c < 1$ that only depends on γ_f and γ_c such that $\delta < e^{-\beta/c}$. Suppose not. Then we have $\delta \geq e^{-\beta/c}$ for every $c < 1$. Therefore, inequality 18 and elementary calculus imply that

$$\gamma_c \geq \beta\zeta + 2e^{-\beta/c} + 1 - \zeta\gamma_f \geq 1 + \zeta(-c\ln\frac{\zeta c}{2} + c - \gamma_f).$$

Now one can choose ζ and $c < 1$ so that right hand side of this inequality is arbitrarily close to $1 + 2e^{-\gamma_f}$, which is a contradiction. Hence at most a $\delta < e^{-\beta/c}$ fraction of cities are indirectly served. As in the proof of Theorem 12, we repeat the algorithm to directly serve these cities too.

Suppose we repeat the algorithm until only 1 indirectly served city remained. Let l be the number of iterations and β_j and δ_j be the β and δ in the jth iteration. We have $|X|\prod_{j=1}^{l}\delta_j = 1$. Therefore,

$$\sum_{j=1}^{l}\beta_j < -c\sum_{j=1}^{l}\ln\delta_j = c\ln|X|.$$

This gives us a set-cover solution within an approximation factor of $c\ln|X|$ for some $c < 1$. $\square$