A new greedy approach for facility location problems

[Extended Abstract]

Kamal Jain
Microsoft Research
One Microsoft Way
Redmond, WA 98052
kamalj@microsoft.com

Mohammad Mahdian
Laboratory for Computer Science
M.I.T.
Cambridge, MA 02139
mahdian@mit.edu

Amin Saberi
College of Computing
Georgia Tech
Atlanta, GA 30332
saberi@cc.gatech.edu

ABSTRACT

We present a simple and natural greedy algorithm for the metric uncapacitated facility location problem achieving an approximation guarantee of 1.61. We use this algorithm to find better approximation algorithms for the capacitated facility location problem with soft capacities and for a common generalization of the k-median and facility location problems. We also prove a lower bound of $1+2/e$ on the approximability of the k-median problem. At the end, we present a discussion about the techniques we have used in the analysis of our algorithm, including a computer-aided method for proving bounds on the approximation factor.

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 facility $i \in \mathcal{F}$ and city $j \in \mathcal{C}$, we have a *connection cost* (a.k.a. service cost) c_{ij} between facility i and city j. 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, 21], and recently in network design problems such as placement of routers and caches [14, 22], agglomeration of traffic or data [1, 15], and web server replications in a content distribution network (CDN) [19, 26]. In the last decade this problem was studied extensively from the perspective of approximation algorithms [2, 4, 6, 7, 8, 13, 18, 20, 28, 30].

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. At the time of submission of the present paper, the best known approximation algorithm for this problem was a 1.728-approximation algorithm due to Charikar and Guha [4]. This algorithm marginally improves an LP-rounding-based algorithm of Chudak and Shmoys [7, 8] using the ideas of cost scaling, greedy augmentations, and a primal-dual algorithm of Jain and Vazirani [18]. The drawback of LP-rounding-based algorithms is that they need to solve large linear programs and therefore have a high running time. Charikar and Guha [4] also present an $O(n^3)$ algorithm with approximation ratio 1.853. Mahdian et al. [23] show that a simple greedy algorithm (similar to the greedy set-cover algorithm of Hochbaum [16]) achieves an approximation ratio of 1.861 in $O(n^2 \log n)$ time. For the case of sparse graphs, Thorup [30] 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 $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. Shmoys [27] provides a 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 algorithm of Mahdian et al [23]. We use the method of *dual fitting* for the analysis of this algorithm. In this method, the algorithm computes a solution to the problem together with an infeasible dual-solution with the same value. The approximation factor of the algorithm can be computed as the factor by which we need to shrink the dual solution to make it feasible. In order to compute this factor, we express the constraints imposed by the problem statement and our algorithm as linear inequalities. This allows us to bound the factor by solving a particular series of linear programs, which we call *factor-revealing LPs*. A more detailed treatment of these techniques will appear in Jain et al [17].

The technique of factor-revealing LPs is similar to the idea of LP bounds in coding theory. LP bounds give the best known bounds on the minimum distance of a code with a given rate by bounding the solution of a linear program. (cf. McEliece et al. [25]). In the context of approximation algorithms, Goemans and Kleinberg [11] use a similar method in the analysis of their algorithm for the minimum latency problem.

The factor-revealing LP enables us to compute 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 approxima-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.
Copyright 2002 ACM 1-58113-495-9/02/0005 ...$5.00.

tion ratio of the 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 algorithms for the facility location problem, the primal-dual algorithm of Jain and Vazirani [18] is perhaps the most versatile one in that it can be used to obtain algorithms for other variants of the problem. This versatility is partly because of a property of the algorithm which makes it possible to apply the Lagrangian relaxation technique. We call this property *the Lagrangian multiplier preserving property*. We will prove in Section 5 that our algorithm also has this property with an approximation factor better than the primal-dual algorithm. This enables us to obtain algorithms for some variants of the facility location problem, such as the k-facility location problem and the capacitated facility location problem with soft capacities. In 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. This problem is a common generalization of the facility location and k-median problems. The k-median problem is studied extensively [2, 4, 5, 18] 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 [18]. In this paper, we present a 4-approximation algorithm for this problem. We will also 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 a 2α-approximation algorithm for the capacitated facility location problem with soft capacities.

In Section 6, we will state some lower bound results. We 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]. This result shows that k-median is a strictly harder problem to approximate than the facility location problem. 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.

In Section 7 we will see a general discussion about the method used to analyze the algorithms in this paper. The important feature of this technique is that the most difficult part of the analysis, which is proving a bound on the solution of the factor-revealing LP, can be done almost automatically using a computer. We will use the set cover problem as an example to illustrate the technique of using factor-revealing LPs.

Since the submission of the present paper, two new algorithms have been proposed for the facility location problem. The first algorithm, due to Sviridenko [29], uses the LP-rounding method to achieve an approximation factor of 1.58. The second algorithm, due to Mahdian, Ye, and Zhang [24], combines our algorithm with the idea of cost scaling to achieve an approximation factor of 1.52.

2. THE ALGORITHM

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.

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

The LP-relaxation of this program is:

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

The dual program is:

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

We can think of the variable α_j in the dual program as the share of city j toward the total expenses. Now, suppose we have an algorithm that 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, $\sum_{j\in S\cap\mathcal{C}} \alpha_j \leq \gamma c_S$, where $\gamma \geq 1$ is a fixed number. Then the approximation ratio of the algorithm is at most γ, since if for every facility i that is opened in the optimal solution and the collection A of cities that are connected to it, we write the inequality $\sum_{j\in A} \alpha_j \leq \gamma(f_i + \sum_{j\in A} c_{ij})$ and add up these inequalities, we will obtain that the cost of our solution is at most γ times the cost of the optimal solution. Another way of looking at this is from the perspective of LP-duality. The inequality $\sum_{j\in S\cap\mathcal{C}} \alpha_j \leq \gamma c_S$ implies that if we shrink α_j's by a factor of γ, we obtain a feasible dual solution. The value of this feasible solution for the dual, which is $\sum_{j\in\mathcal{C}} \alpha_j/\gamma = T/\gamma$, is a lower bound on the cost of the optimum.

This method, which is called *dual fitting*, can be considered a primal-dual type method. The only difference is that in primal-dual algorithms, we usually relax the complementary slackness conditions to obtain a solution for the primal and a solution for the dual so that the ratio of the values of the objective functions for these two solutions is bounded by the approximation factor of the algorithm. However, in the dual fitting scheme we relax the inequalities in the dual program. Therefore, the algorithm find a solution for the primal, and an infeasible solution for the dual with the same value for the objective function. The amount by which the dual inequalities are relaxed (or in other words, the amount by which we must shrink the dual solution so that it *fits* the dual) will give a bound on the approximation factor of the algorithm. This fact is the basis of our analysis. See Jain et al. [17] or Mahdian et al. [23] for a more detailed discussion of this technique.

Algorithm 1

1. We introduce a notion of time. The algorithm starts at time 0. At this time, all cities are unconnected, all facilities are unopened, 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 *unopened* 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 amount j would save by switching 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 unopened 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.

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.

At any time during the execution of this algorithm, the budget of each connected city is equal to its current connection cost plus its total contribution toward open facilities. 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.*

The above algorithm is similar to the greedy algorithm of Mahdian et al [23]. The only difference is that in [23], cities stop offering money to facilities as soon as they get connected to a facility, but in our algorithm, they still offer some money (the amount that they could save by switching their facility) to other facilities. As a result, our algorithm finds a solution that cannot be improved just by opening new facilities, and therefore it cannot be improved by the greedy augmentation procedure of Charikar and Guha [4], whereas the solution found by the algorithm of Mahdian et al. [23] does not possess this property. As we will see in the next section, this change reduces the approximation factor of the algorithm from 1.86 to 1.61.

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-revealing LP) whose solution gives the value of γ. In Section 3.2 we will use the factor-revealing LP to prove an upper bound of 1.61 on the approximation ratio of Algorithm 1. A discussion of this technique is presented in Section 7.

3.1 Deriving the factor-revealing 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 $r_{j,i} := \alpha_j$. 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 constant and it cannot revoke' 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 time $t = \alpha_i - \epsilon$. At this time, the amount city j offers to 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 by the definition of $r_{j,i}$ 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, for all i,

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

The triangle inequality is another important constraint that we need to use. Consider cities i and j with $j < i$ at time $t = \alpha_i - \epsilon$. Let f' be the facility j is connected to at time t. By the triangle inequality and the definition of $r_{j,i}$, the connection cost $c_{f'i}$ between city i and facility f' is at most $r_{j,i} + d_i + d_j$. Furthermore, $c_{f'i}$ can not 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 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. Therefore for every $1 \leq j < i \leq k$,

$$\alpha_i \leq r_{j,i} + d_i + d_j. \quad (7)$$

The above inequalities form the following optimization program, which we call the factor-revealing LP.

$$\begin{array}{ll} \text{maximize} & \dfrac{\sum_{i=1}^{k} \alpha_i}{f + \sum_{i=1}^{k} d_i} \qquad (8) \\ \text{subject to} & \forall\, 1 \leq i < k: \ \alpha_i \leq \alpha_{i+1} \\ & \forall\, 1 \leq j < i < k: \ r_{j,i} \geq r_{j,i+1} \\ & \forall\, 1 \leq j < i \leq k: \ \alpha_i \leq r_{j,i} + d_i + d_j \\ & \forall\, 1 \leq i \leq k: \ \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \\ & \qquad + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f \\ & \forall\, 1 \leq j \leq i \leq k: \ \alpha_j, d_j, f, r_{j,i} \geq 0 \end{array}$$

Notice that although the above 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.

Lemma 2. *If z_k denotes the solution of the factor-revealing 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-revealing LP. Thus, the value of the objective function for this solution is at most z_k. □

Lemmas 1 and 2 imply the following.

k	$\max_{i<k} z_i$
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-revealing LP

Lemma 3. *Let z_k be the solution of the factor-revealing LP, and $\gamma := \sup_k\{z_k\}$. Then Algorithm 1 solves the metric facility location problem with an approximation factor of γ.*

3.2 Solving the factor-revealing 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-revealing LP for small values of k, in order to compute the numerical value of γ. Table 1 shows a summary of results that are obtained by solving the factor-revealing LP using CPLEX. 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-revealing 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-revealing 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-revealing 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. We have used this technique to prove an upper bound of 1.61 on γ. The proof of this upper bound is presented in Appendix A. Also, we can use the optimal solution of the factor-revealing LP to construct an example on which our algorithm performs at least z_k times worse than the optimum. The proof of this fact is omitted here. These results imply the following.

Theorem 4. *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 5. *Let $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program.*

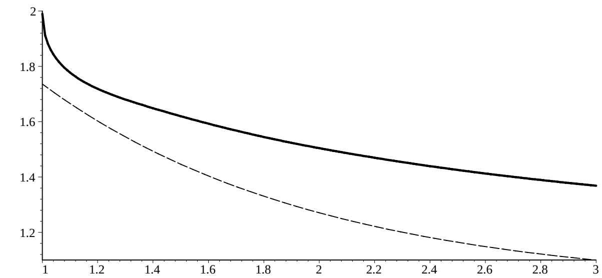

Figure 1: The tradeoff between γ_f and γ_c

$$
\begin{aligned}
\textit{maximize} \quad & \frac{\sum_{i=1}^{k} \alpha_i - \gamma_f f}{\sum_{i=1}^{k} d_i} && (9)\\
\textit{subject to} \quad & \forall\, 1 \le i < k: \ \alpha_i \le \alpha_{i+1} \\
& \forall\, 1 \le j < i < k: \ r_{j,i} \ge r_{j,i+1} \\
& \forall\, 1 \le j < i \le k: \ \alpha_i \le r_{j,i} + d_i + d_j \\
& \forall\, 1 \le i \le k: \ \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \\
& \qquad\qquad + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f \\
& \forall\, 1 \le j \le i \le k: \ \alpha_j, d_j, f, r_{j,i} \ge 0
\end{aligned}
$$

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 stated in Section 6. Similar tradeoff problems are considered by Charikar and Guha [4]. However, an important advantage that we get here is that all the inequalities $ALG \le \gamma_f F_{SOL} + \gamma_c C_{SOL}$ are satisfied by a *single* algorithm. In the next section, we will use the point $\gamma_f = 1$ of this tradeoff to design algorithms for other variants of the facility location problem. Other points of this tradeoff can also be useful in designing other algorithms based on our algorithm. For example, Mahdian, Ye, and Zhang [24] use the point $\gamma_f = 1.1$ of this tradeoff to obtain a 1.52-approximation algorithm for the metric facility location problem, which is currently the best known algorithm for this problem.

5. VARIANTS OF THE 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. The *k-facility location problem* is a common generalization of k-median and the facility location problem. In this problem, we have an upper bound k on the number of facilities that can be opened, as well as costs for opening facilities. Jain and Vazirani [18] reduced the k-median 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 $\mathcal{I}$, $C \le \alpha(OPT - F)$. Jain and Vazirani [18] 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. They have noted that this result also holds for the k-facility location problem.

LEMMA 6. **[18]** *An LMP α-approximation algorithm for the facility location problem gives a 2α-approximation algorithm for the k-facility location problem.*

In this section, we give an LMP 2-approximation algorithm for the metric facility location problem based on Algorithm 1. This will result in a 4-approximation algorithm for the metric k-facility location problem, whereas the best previously known was 6 [18].

In the capacitated facility location problem, for every facility, there is one more parameter, 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.* Jain and Vazirani [18] show that their facility location algorithm gives rise to a 4-approximation algorithm for the metric capacitated facility location problem with soft capacities. One can easily generalize their result to the following lemma. This lemma, together with our LMP 2-approximation facility location algorithm gives a 3-approximation algorithm for the metric capacitated facility location problem with soft capacities.

LEMMA 7. *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.*

Now we show that there is an LMP 2-approximation algorithm for the metric facility location problem. The proof is based on Theorem 5 together with the scaling technique of Charikar and Guha [4]. We prove the following lemma using this technique.

LEMMA 8. *Assume there is an algorithm $\mathcal{A}$ for the metric facility location problem such 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 $\mathcal{A}$ on this modified instance $\mathcal{I}'$, and outputs its answer. It is easy to see that this algorithm is an LMP α-approximation. $\square$

Now we only need to prove the following. The proof of this theorem follows the general scheme that is explained in Section 7.

THEOREM 9. *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.*

PROOF. By Theorem 5 we only need to prove that the solution of the factor-revealing LP 9 with $\gamma_f = 1$ is at most 2. We first write the maximization program 9 as the following equivalent linear program.

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} \alpha_i - f \qquad\qquad (10)\\
\text{subject to} \quad & \sum_{i=1}^{k} d_i = 1\\
& \forall\, 1 \le i < k: \ \alpha_i - \alpha_{i+1} \le 0\\
& \forall\, 1 \le j < i < k: \ r_{j,i+1} - r_{j,i} \le 0\\
& \forall\, 1 \le j < i \le k: \ \alpha_i - r_{j,i} - d_i - d_j \le 0\\
& \forall\, 1 \le j < i \le k: \ r_{j,i} - d_i - g_{i,j} \le 0\\
& \forall\, 1 \le i \le j \le k: \ \alpha_i - d_j - h_{i,j} \le 0\\
& \forall\, 1 \le i \le k: \ \sum_{j=1}^{i-1} g_{i,j} + \sum_{j=i}^{k} h_{i,j} - f \le 0\\
& \forall\, i,j: \ \alpha_j, d_j, f, r_{j,i}, g_{i,j}, h_{i,j} \ge 0
\end{aligned}$$

We need to prove an upper bound of 2 on the solution of the above LP. Since this program is a maximization program, it is enough to prove the upper bound for any relaxation of the above program. Numerical results (for a fixed value of k, say $k = 100$) suggest that removing the second, third, and seventh inequalities of the above program does not its solution. Therefore, we can relax the above program by removing these inequalities. Now, it is a simple exercise to write down the dual of the relaxed linear program and compute its optimal solution. This solution corresponds to multiplying the third, fourth, fifth, and sixth inequalities of the linear program 10 by $1/k$, and the first one by $(2-1/k)$, and adding up these inequalities. This gives an upper bound of $2 - 1/k$ on the value of the objective function. Thus, for $\gamma_f = 1$, we have $\gamma_c \le 2$. In fact, γ_c is precisely equal to 2, as shown by the following solution for the program 9.

$$\alpha_i = \begin{cases} 2 - \frac{1}{k} & i = 1 \\ 2 & 2 \le i \le k \end{cases}$$

$$d_i = \begin{cases} 1 & i = 1 \\ 0 & 2 \le i \le k \end{cases}$$

$$r_{j,i} = \begin{cases} 1 & j = 1 \\ 2 & 2 \le j \le k \end{cases}$$

$$f = 2(k-1)$$

This example shows that the above analysis of the factor-revealing LP is tight. □

Lemma 8 and Theorem 9 provide an LMP 2-approximation algorithm for the metric facility location problem. This result improves all the results in Jain and Vazirani [18], 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. Our first result is the following theorem, which together with Feige's result on the hardness of set-cover [10] shows that there is no $(1+\frac{2}{e}-\epsilon)$-approximation algorithm for k-median, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. The proof is similar to the one used by Guha and Khuller [13] to prove the hardness of the metric facility location problem, and is omitted in this extended abstract.

THEOREM 10. *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 $c < 1$.*

This theorem improves a lower bound of $1+\frac{1}{e}$ due to Guha [12]. Notice that the above theorem 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.

We also adapt the proof of Charikar and Guha [13] to show the following lower bound on the tradeoff results. The dashed line in Figure 1 shows the lower bound provided by the following theorem.

THEOREM 11. *Let γ_f and γ_c be constants with $\gamma_c < 1 + 2e^{-\gamma_f}$. Assume there is an algorithm $\mathcal{A}$ such that for every instance $\mathcal{I}$ of the metric facility location problem, $\mathcal{A}$ 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 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 found by Guha [12] show that Lemma 6 is tight. This shows that one cannot use Lemma 6 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 by Charikar and Guha [4].

7. THE FACTOR-REVEALING LP TECHNIQUE

In this section, we elaborate on the technique of using factor-revealing LPs which we used to analyze the algorithms in this paper. We demonstrate this technique by applying it in combination with dual fitting to a classical greedy algorithm for the set cover problem. We also explain how we can use computers to predict and prove bounds on the solution to the factor-revealing LP. Similar methods are used in Mahdian et al. [23] and Goemans and Kleinberg [13].

A re-statement of the greedy algorithm for the set cover problem is as follows. All uncovered elements raise their dual-variables until a new set S goes tight (i.e., its cost equals the sum of the values of the dual variables of its elements). At this point, the set S is picked. Newly covered elements pay for the cost of S with their dual values. In doing so, they withdraw their contributions offered towards the cost of any other set. This ensures that at the end of the algorithm the total contribution of the elements is equal to the sum of the cost of the picked sets. However, we might not get a feasible dual solution. To make the dual solution feasible we look for the smallest positive number Z, so that when the dual solution is shrunk by a factor of Z, it becomes feasible. An upper bound on the approximation factor of the algorithm is obtained by maximizing Z over all possible instances. This technique is called *dual fitting* and is explained in detail in Mahdian et al [23]. In this section, we focus on the *factor-revealing LP* technique, which is used to estimate the value of Z.

Clearly Z is also the maximum factor by which any set is over-tight. Consider any set S. We want to see what is the worst factor, over all sets and over all possible instances of the problem, by which a set S is over-tight. Let the elements in S be $1, 2, \cdots, k$. Let x_i be the dual variable corresponding to the element i at the end of the algorithm. Without loss of generality we may assume that $x_1 \le x_2 \le \cdots \le x_k$. It is easy to see that at time $t = x_i^-$, total duals offered to S is at least $(k-i+1)x_i$. Therefore, this value cannot be greater than the cost of the set S (denoted by c_S). So, the optimum solution of the following mathematical program gives an upper bound on the value of Z. (Note that c_S is a variable not a constant).

$$\begin{aligned} \text{maximize} \quad & \frac{\sum_{i=1}^{k} x_i}{c_S} && (11) \\ \text{subject to} \quad & \forall 1 \le i < k: \ x_i \le x_{i+1} \\ & \forall 1 \le i \le k: \ (k-i+1)x_i \le c_S \\ & \forall 1 \le i \le k: \ x_i \ge 0 \\ & c_S \ge 1 \end{aligned}$$

The above optimization program can be turned into a linear program by adding the constraint $c_S = 1$ and changing the objective function to $\sum_{i=1}^{k} x_i$. We call this linear program the *factor-revealing LP*. Notice that the factor-revealing LP has nothing to do with the LP formulation of the set cover problem; it is only used in order to analyze this particular algorithm. This is the important distinction between the factor-revealing LP technique, and other LP-based techniques in approximation algorithms.

Once we formulate the analysis of the algorithm as a factor-revealing LP, we can use computers to empirically compute the upper bound given by the factor-revealing LP on the approximation ratio of the algorithm. This is very useful, since if the empirical results suggest that the factor-revealing LP does not give us a good approximation ratio, we can try adding other inequalities to the factor-revealing LP. For this we might need to introduce new variables to capture the execution of the algorithm more accurately, e.g., we needed to introduce the variables $r_{j,i}$ in Section 3.1 in order to get a good bound on the approximation ratio of the algorithm.

The next step is to analyze the factor-revealing LP and derive an upper bound on the value of its solution. For the set cover example above, this step is trivial, since the factor-revealing LP associated with the algorithm is quite simple. However, in general this can be the most difficult step of the proof (as it is in the case of our algorithm and the algorithm of Mahdian et al. [23]). Here we can use computers to get ideas about the proof, as explained below. Proving Theorem 4 would have been very difficult without using these techniques.

Since the factor-revealing LP provides an upper bound on the approximation ratio of the algorithm, we can relax some of the constraints of this LP to make it simpler. After each relaxation, we can use computers to verify that this relaxation does not change the value of the objective function drastically. After simplifying the factor-revealing LP in this way, we can find an upper bound on its solution by finding a feasible solution for its dual *for every k*. Again, here we can use a computer to solve the dual linear program for a couple hundred values of k, to observe a trend in the values of the optimal dual solution. After guessing a sequence of dual solutions, one has to theoretically verify their feasibility. For complicated linear programs, it is usually a good idea to throw in a few parameters (like p_1 and p_2 in the proof of Theorem 4 in Appendix A), guess a general dual solution in terms of these parameters, and optimize over the choice of these parameters at the end.

Note that in general this technique does not guarantee the tightness of the analysis, because sometimes the algorithm performs well not because of local structures but for some global reasons. Still, in many cases one may get a tight example from a feasible solution of the factor-revealing LP. For example, from any feasible solution x of the factor-revealing LP 11, one can construct the following instance: There are k elements $1, \ldots, k$, a set $S = \{1, \ldots, k\}$ of cost $1 + \epsilon$ which is the optimal solution, and sets $S_i = \{i\}$ of cost x_i for $i = 1, \ldots, k$. It is easy to verify that our algorithm works $\sum x_i$ times worst than the optimal on this instance. This means that the approximation ratio of the set cover algorithm is precisely equal to the solution of the factor-revealing LP, which is H_n.

8. CONCLUDING REMARKS

A large fraction of the theory of approximation algorithms, as we know it today, is built around the theory of linear programming, which offers the two fundamental algorithm design techniques of rounding and the primal-dual schema (see Vazirani [31]). The technique of using dual fitting with the factor-revealing LP appears to be a third emerging technique.

The technique that we used in this paper seems to be a useful tool for analyzing greedy, heuristic, and local search algorithms. For many algorithms, the proof of the approximation ratio is mainly based on combining several inequalities (usually linear inequalities) to derive a bound on the approximation ratio. It might be possible to "automatize" such proofs using a method similar to the one used in this paper. It would be interesting to find other examples that apply 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 to 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. Section 7 was developed over time, while the last two authors were also working on another paper [23] with Vijay V. Vazirani and Evangelos Markakis. We would like to thank Vijay Vazirani, Evangelos Markakis, Michel Goemans, Nicole Immorlica, and Milena Mihail for their helpful comments.

9. 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, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. Approximation algorithms for facility location via dual fitting with factor-revealing lp. submitted to Combinatorica.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. Unpublished manuscript, 2002.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] M. Sviridenko. An 1.582-approximation algorithm for the metric uncapacitated facility location problem. to appear in the Ninth Conference on Integer Programming and Combinatorial Optimization, 2002.

[30] 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.

[31] V. V. Vazirani. *Approximation Algorithms.* Springer-Verlag, Berlin, 2001.

APPENDIX

A. UPPER BOUND ON THE SOLUTION OF THE FACTOR-REVEALING LP

In this appendix we prove an upper bound of 1.61 on the solution of the factor-revealing LP 8. This proof is obtained using the techniques explained in Section 7. We start by proving the following lemma which allows us to concentrate on the case when k is sufficiently large.

LEMMA 12. *If z_k denotes the solution to the factor-revealing 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-revealing 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, $z_{2k} \ge z_k$. □

LEMMA 13. *Let z_k be the solution to the factor-revealing LP. Then for every sufficiently large k, $z_k \le 1.61$.*

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

$$(i' - i + 1)\alpha_i \le \sum_{j=i}^{i'} d_j + f - \sum_{j=1}^{i-1} x_{j,i}. \qquad (12)$$

Now, we define l_i as follows:

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

where p_1 and p_2 are two constants (with $p_1 < p_2$) that will be fixed later. Consider Inequality 12 for every $i \le p_2 k$ and $i' = l_i$, and divide both sides of this inequality by $(l_i - i + 1)$. By adding up these inequalities we obtain

$$\sum_{i=1}^{p_2 k} \alpha_i \le \sum_{i=1}^{p_2 k} \sum_{j=i}^{l_i} \frac{d_j}{l_i - i + 1} + (\sum_{i=1}^{p_2 k} \frac{1}{l_i - i + 1})f - \sum_{i=1}^{p_2 k} \sum_{j=1}^{i-1} \frac{x_{j,i}}{l_i - i + 1}. \qquad (13)$$

Now for every $j \le p_2 k$, let $y_j := x_{j,p_2 k}$. The second inequality of the factor-revealing LP implies that $x_{j,i} \ge y_j$ for every $j < i \le p_2 k$ and $x_{j,i} \le y_j$ for every $i > p_2 k$. Also, let $\zeta := \sum_{i=1}^{p_2 k} \frac{1}{l_i - i + 1}$. Therefore, inequality 13 implies

$$\sum_{i=1}^{p_2 k} \alpha_i \le \sum_{i=1}^{p_2 k} \sum_{j=i}^{l_i} \frac{d_j}{l_i - i + 1} + \zeta f - \sum_{i=1}^{p_2 k} \sum_{j=1}^{i-1} \frac{y_j}{l_i - i + 1}. \quad (14)$$

Consider the index $\ell \le p_2 k$ for which $2d_\ell + y_\ell$ has its minimum (i.e., for every $j \le p_2 k$, $2d_\ell + y_\ell \le 2d_j + y_j$). The third inequality of the factor-revealing LP implies that for $i = p_2 k + 1, \ldots, k$,

$$\alpha_i \le r_{\ell,i} + d_i + d_\ell \le x_{\ell,i} + 2d_\ell + d_i \le d_i + 2d_\ell + y_\ell. \quad (15)$$

By adding Inequality 15 for $i = p_2 k + 1, \ldots, k$ with Inequality 14 we obtain

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

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 $j \le p_2 k$. Now, let $\zeta' := 1 + \sum_{i=p_1 k+1}^{p_2 k} \frac{1}{k-i+1}$ and $\delta := (1 - p_2) - \frac{1}{2k} \sum_{j=1}^{p_2 k} \sum_{i=j+1}^{p_2 k} \frac{1}{l_i - i + 1}$. Therefore, the above inequality can be written as follows:

$$\sum_{i=1}^{k} \alpha_i \le \sum_{j=1}^{p_2 k} \zeta d_j + \sum_{j=p_2 k+1}^{k} \zeta' d_j + \zeta f + \delta(2d_\ell + y_\ell)k, \quad (16)$$

where

$$\zeta = \sum_{i=1}^{p_2 k} \frac{1}{l_i - i + 1} = \ln \frac{p_2(1-p_1)}{(p_2-p_1)(1-p_2)} + o(1), \tag{17}$$

$$\zeta' = 1 + \sum_{i=p_1 k+1}^{p_2 k} \frac{1}{k - i + 1} = 1 + \ln \frac{1-p_1}{1-p_2} + o(1), \tag{18}$$

$$\begin{aligned} \delta &= 1 - p_2 - \frac{1}{2k} \sum_{j=1}^{p_2 k} \sum_{i=j+1}^{p_2 k} \frac{1}{l_i - i + 1} \\ &= \frac{1}{2}(2 - p_2 - p_2 \ln \frac{p_2}{p_2 - p_1} - \ln \frac{1-p_1}{1-p_2}) + o(1). \end{aligned} \tag{19}$$

Now if we choose p_1 and p_2 such that $\delta < 0$, and let $\gamma := \max(\zeta, \zeta')$ then inequality 16 implies that

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

Using equations 17, 18, and 19, 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 $\gamma < 1.61$. $\square$