Improved Approximation Algorithms for Metric Facility Location Problems

Mohammad Mahdian* Yinyu Ye† Jiawei Zhang ‡

May 9, 2002

Abstract

In this paper we present a 1.52-approximation algorithm for the uncapacitated metric facility location problem. This algorithm uses the idea of cost scaling, a greedy algorithm of Jain, Mahdian, and Saberi, and a greedy augmentation procedure of Charikar, Guha, and Khuller. We also present a 2.89-approximation for the capacitated metric facility location problem with soft capacities.

1 Introduction

In the uncapacitated facility location problem (UFLP), 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.

The facility location problem is a central problem in operations research, and a large number of approximation algorithms using a variety of techniques have been proposed for this problem. Table 1 shows a summary of the results. The running times in this table are in terms of $n = n_f + n_c$. Regarding hardness results, Guha and Khuller [4] proved that it is impossible to get an approximation guarantee of 1.463 for the uncapacitated metric facility location problem, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log \log n)}]$. For a more detailed survey on this problem, see Shmoys [10].

In this paper, we combine the greedy algorithm of Jain, Mahdian, and Saberi [6] and the greedy augmentation of Charikar, Guha, and Khuller [1, 4] to show that UFLP can be approximated within a factor of 1.52, whereas the best previously known factor was 1.582 [12]. Note that this approximation factor is very close to the lower bound of 1.463 proved by Guha and Khuller [4].

*Laboratory for Computer Science, MIT, Cambridge, MA 02139, USA. E-mail: mahdian@mit.edu.

†Department of Management Science and Engineering, School of Engineering, Stanford University, Stanford, CA 94305, USA. E-mail: yinyu-ye@stanford.edu. Research supported in part by NSF grants DMI-9908077.

‡Department of Mamagement Sciences, Henry B. Tippie College of Business, The University of Iiowa, Iowa City, IA 52242, USA. E-mail: jiawei-zhang@stanford.edu. Research supported in part by NSF grants DMI-9908077.

approx. factor	reference	technique/running time
$O(\ln n_c)$	Hochbaum [5]	greedy algorithm/$O(n^3)$
3.16	Shmoys et al. [11]	LP rounding
2.41	Guha and Khuller [4]	LP rounding + greedy augmentation
1.736	Chudak [2]	LP rounding
$5+\epsilon$	Korupolu et al. [8]	local search/$O(n^6 \log(n/\epsilon))$
3	Jain and Vazirani [7]	primal-dual method/$O(n^2 \log n)$
1.853	Charikar and Guha [1]	primal-dual method + greedy augmentation/$O(n^3)$
1.728	Charikar and Guha [1]	LP rounding + primal-dual method + greedy augmentation
1.861	Mahdian et al. [9]	greedy algorithm/$O(n^2 \log n)$
1.61	Jain et al. [6]	greedy algorithm/$O(n^3)$
1.582	Sviridenko [12]	LP rounding
1.52	This paper	greedy algorithm + greedy augmentation/$O(n^3)$

Table 1: Approximation Algorithms for UFLP

As a by-product of our analysis, we also present an improved 2.89-approximation algorithm for the capacitated facility location problem (CFLP) where opening multiple copies of the same facility is allowed. The best previously known factor was 3 [6]. Also, Chudak and Shmoys [3] gave a 3-approximation algorithm for CFLP with uniform capacities.

The algorithm for UFLP and its underlying intuition are presented in Section 2. In Section 3, we prove the upper bound of 1.52 on the approximation factor of the algorithm for UFLP. In Section 4, we present the 2.89-approximation for CFLP. Some possible directions for the future research on these problems are discussed in Section 5.

2 Algorithm

Jain, Mahdian, and Saberi [6] proposed a greedy algorithm for UFLP that achieves a factor of 1.61. Here is a sketch of their algorithm:

The JMS Algorithm

1. At the beginning, 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 equal to $\max(B_j - c_{ij}, 0)$ if j is unconnected, and $\max(c_{i'j} - c_{ij}, 0)$ if it is connected to some other facility i'.

2. While there is an unconnected city, increase the budget of each *unconnected* city at the same rate, until one of the following events occurs:

   (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.

   (b) For some unconnected city j, and some facility i that is already open, the budget of j is equal to the connection cost c_{ij}. In this case, we connect j to i.

One important property of the above algorithm is that it finds a solution in which there is no unopened facility that one can open to decrease the cost (without closing any other facility). This is because for each city j and facility i, j offers to i the amount that it would save in the connection cost if it gets its service from i. This is, in fact, the main advantage of the JMS algorithm over a previous algorithm of Mahdian et al. [9].

Here we use the JMS algorithm to solve UFLP with an improved approximation factor. Our algorithm has two phases. In the *first* phase, we scale up the opening costs of all facilities by a factor of δ (which is a constant that will be fixed later) and then run the JMS algorithm to find a solution SOL_1. The technique of cost scaling has been previously used by Charikar and Guha [1] for the facility location problem, in order to take advantage of the asymmetry between the performance of the algorithm with respect to facility and connection costs. Here we use this idea for a different reason: Intuitively, facilities that are opened by the JMS algorithm with scaled-up facility costs are those that are very economical, because we weight the facility cost more than the connection cost in the objective function. Therefore, we open these facilities in the first phase of the algorithm.

In the *second* phase of the algorithm, we scale down the opening costs of facilities back to their original values all at the same rate. If at any point during this process, a facility could be opened without increasing the total cost (i.e., if the the opening cost of the facility equals the total amount that cities can save by switching their "service provider" to that facility), then we open the facility and connect each city to its closest open facility. It is not difficult to see that this is equivalent to a greedy procedure introduced by Guha and Khuller [4] and Charikar and Guha [1]. In this procedure, in each iteration, we pick a facility u of opening cost f_u such that if by opening u, the total connection cost decreases from C to C', the ratio $(C - C' - f_u)/f_u$ is maximized. If this ratio is positive, then we open the facility u, and iterate; otherwise we stop.

Let SOL_2 denote the solution after the above greedy augmentation procedure. We will prove in the next section that the cost of SOL_2 is at most 1.52 times the cost of the optimal solution.

3 The approximation factor

In order to analyze the approximation factor of our algorithm, we use results of [6] and [1] that bound the cost of the solution found by the JMS algorithm and the greedy augmentation procedure.

Before we present the result of the [6], we give the following definition.

Definition 1 *An algorithm is called a (γ_f, γ_c)-approximation algorithm for UFLP, if for every instance $\mathcal{I}$ of UFLP, 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 the algorithm is at most $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.*

The following theorem gives tight bounds on the tradeoff between γ_f, γ_c for the JMS algorithm.

Lemma 1 ([6]) *Let $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program (which we call the factor-revealing LP).*

$$\text{maximize} \quad \frac{\sum_{i=1}^{k} \alpha_i - \gamma_f f}{\sum_{i=1}^{k} d_i} \tag{1}$$

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

Then the JMS algorithm is a (γ_f, γ_c)-approximation algorithm for UFLP.

In particular, it is proved [6] that $\gamma_c \le 1.61$ when $\gamma_f = 1.61$, and therefore the JMS algorithm is a (1.61, 1.61)-approximation algorithm. Here we use the above theorem with $\gamma_f = 1.11$. The following lemma shows that JMS algorithm is a $(1.11, 1.78)$-approximation algorithm for UFLP. The proof is long and technical and is presented in Appendix A.

Lemma 2 *For every k, the solution of the maximization program (1) with $\gamma_f = 1.11$ is at most 1.78. In other words, $\gamma_c \le 1.78$ when $\gamma_f = 1.11$.*

We also use the following result of Charikar and Guha [1] that bounds the cost of the solution after running the greedy augmentation procedure in terms of the cost of the initial solution and an arbitrary solution.

Lemma 3 ([1]) *For every instance $\mathcal{I}$ of UFLP and for every solution SOL of $\mathcal{I}$ with facility cost F_{SOL} and connection cost C_{SOL}, if an initial solution has facility cost F and connection cost C, then after greedy augmentation the cost of the solution is at most*

$$F + F_{SOL} \max\left\{0, \ln\left(\frac{C - C_{SOL}}{F_{SOL}}\right)\right\} + F_{SOL} + C_{SOL}.$$

Using the above lemmas, we can prove the following.

Theorem 4 *The uncapacitated facility location problem can be approximated within a factor of 1.52 in time $O(n^3)$.*

Proof : Let OPT be an optimal solution with facility and connection costs F^* and C^*, respectively, and consider a pair (γ_f, γ_c) given in Lemma 1. Let SOL_1 denote the solution found by the JMS algorithm for an instance in which facility costs are scaled by a factor of δ ($\delta \ge 1$). By Lemma 1, the cost of this solution, evaluated using scaled-up facility costs, is at most $\gamma_f \delta F^* + \gamma_c C^*$. Therefore, if F_{SOL_1} and C_{SOL_1} denote the facility and connection costs of SOL_1, evaluated with the original costs, then we have

$$\delta F_{SOL_1} + C_{SOL_1} \le \gamma_f \delta F^* + \gamma_c C^*. \tag{2}$$

Also, by Lemma 3 the cost of the solution returned by the greedy augmentation procedure is at most

$$\text{cost}(SOL_2) \le F_{SOL_1} + F^* \max\left\{0, \ln\left(\frac{C_{SOL_1} - C^*}{F^*}\right)\right\} + F^* + C^* \tag{3}$$

Now, we consider two cases based on whether $C_{SOL_1} < F^* + C^*$ or $C_{SOL_1} \geq F^* + C^*$. In the first case, using inequality (2) we have

$$\begin{aligned} F_{SOL_1} + C_{SOL_1} &= \frac{\delta F_{SOL_1} + C_{SOL_1}}{\delta} + \left(1 - \frac{1}{\delta}\right) C_{SOL_1} \\ &\leq \frac{\gamma_f \delta F^* + \gamma_c C^*}{\delta} + \left(1 - \frac{1}{\delta}\right)(F^* + C^*) \\ &= \left(\gamma_f + 1 - \frac{1}{\delta}\right) F^* + \left(1 + \frac{\gamma_c - 1}{\delta}\right) C^*. \end{aligned} \tag{4}$$

Therefore, since the greedy augmentation procedure never increases the cost, the cost of the final solution SOL_2 of our algorithm is at most

$$\text{cost}(SOL_2) \leq \max\left(\gamma_f + 1 - \frac{1}{\delta}, 1 + \frac{\gamma_c - 1}{\delta}\right) \text{cost}(OPT). \tag{5}$$

In the second case ($C_{SOL_1} \geq F^* + C^*$), by inequality (2) we have

$$C_{SOL_1} \leq \gamma_f \delta F^* + \gamma_c C^* - \delta F_{SOL_1}. \tag{6}$$

Also, since $C_{SOL_1} \geq F^* + C^*$, we have $\ln\left(\frac{C_{SOL_1} - C^*}{F^*}\right) \geq 0$. Therefore, by inequalities (3) and (6) we have

$$\begin{aligned} \text{cost}(SOL_2) &\leq F_{SOL_1} + F^* \ln\left(\frac{C_{SOL_1} - C^*}{F^*}\right) + F^* + C^* \\ &\leq F_{SOL_1} + F^* \ln\left(\frac{\gamma_f \delta F^* + (\gamma_c - 1)C^* - \delta F_{SOL_1}}{F^*}\right) + F^* + C^* \end{aligned} \tag{7}$$

Considering F_{SOL_1} as a variable while all others are fixed, we have the above term maximized at $F_{SOL_1} = (\gamma_f - 1)F^* + \frac{\gamma_c - 1}{\delta} C^*$. Therefore,

$$\begin{aligned} \text{cost}(SOL_2) &\leq (\gamma_f + \ln \delta) F^* + (1 + \frac{\gamma_c - 1}{\delta}) C^* \\ &\leq \max\left(\gamma_f + \ln \delta, 1 + \frac{\gamma_c - 1}{\delta}\right) \text{cost}(OPT) \end{aligned} \tag{8}$$

Inequalities (5) and (8) show that in either case, our algorithm finds a solution whose cost is at most a factor of $\alpha := \max\left(\gamma_f + \ln \delta, 1 + \frac{\gamma_c - 1}{\delta}, \gamma_f + 1 - \frac{1}{\delta}\right)$ more than the optimal solution. By Lemma 2, we can pick $(\gamma_f, \gamma_c) = (1.11, 1.78)$. By minimizing α over the choice of δ, we obtain $\delta = 1.504$ and $\alpha \approx 1.519 < 1.52$. Therefore, our algorithm is a 1.52-approximation algorithm for the uncapacitated facility location problem. It is easy to see that this algorithm can be implemented in $O(n^3)$ time. □

4 Capacitated facility location with soft capacities

In this section, we present a 2.89-approximation algorithm for the metric capacitated facility location problem (CFLP) with soft capacities. CFLP is similar to UFLP except that there is a *capacity* u_i associated with each facility i, that means that facility i can only serve at most u_i cities. This problem has two variants: CFLP with soft capacities, and CFLP with hard capacities. In CFLP with soft capacities, it is allowed to open multiple copies of each facility, while in CFLP with hard capacities each facility can be opened at most once. In this section, we will only talk about the variant with soft capacities.

More precisely, CFLP can be formulated by the following integer program.

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} c_{ij} x_{ij} \\ \text{subject to} \quad & \forall\, i \in \mathcal{F}, j \in \mathcal{C}: \; x_{ij} \le y_i \\ & \forall\, i \in \mathcal{F}: \; \sum_{j\in\mathcal{C}} x_{ij} \le u_i y_i \\ & \forall\, j \in \mathcal{C}: \; \sum_{i\in\mathcal{F}} x_{ij} = 1 \\ & \forall\, i \in \mathcal{F}, j \in \mathcal{C}: \; x_{ij} \in \{0,1\} \\ & \forall\, i \in \mathcal{F}: \; y_i \text{ is a nonnegative integer} \end{aligned} \tag{9}$$

It is known that using the Lagrangian relaxation technique of Jain and Vazirani [7] one can get a 2γ-approximation algorithm for CFLP from an γ-approximation algorithm for the UFLP; and an $(\gamma+1)$-approximation algorithm for CFLP from a $(1,\gamma)$-approximation algorithm for UFLP. Our next theorem generalizes the above results.

Theorem 5 *Any (γ_f, γ_c)-approximation algorithm for UFLP implies a $(\gamma_f + \gamma_c)$-approximation algorithm for CFLP with soft capacities.*

Proof : Let $\lambda = \frac{\gamma_c}{\gamma_f+\gamma_c} \in [0,1]$. For any solution (x_{ij}, y_i) of the integer program (9), the inequality $\sum_{j\in\mathcal{C}} x_{ij} \le u_i y_i$ implies that $\sum_{i\in\mathcal{F}} (1-\lambda) f_i y_i + \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} (c_{ij} + \lambda \cdot \frac{f_i}{u_i}) x_{ij} \le \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} c_{ij} x_{ij}$. Therefore, the following integer program is a relaxation of the integer program (9), i.e., its solution is a lower bound on the solution of (9).

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in\mathcal{F}} (1-\lambda) f_i y_i + \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} (c_{ij} + \lambda \cdot \frac{f_i}{u_i}) x_{ij} \\ \text{subject to} \quad & \forall\, i \in \mathcal{F}, j \in \mathcal{C}: \; x_{ij} \le y_i \\ & \forall\, j \in \mathcal{C}: \; \sum_{i\in\mathcal{F}} x_{ij} = 1 \\ & \forall\, i \in \mathcal{F}, j \in \mathcal{C}: \; x_{ij} \in \{0,1\} \\ & \forall\, i \in \mathcal{F}: \; y_i \text{ is a nonnegative integer} \end{aligned} \tag{10}$$

It is easy to see that this relaxation is an uncapacitated facility location problem where the connection cost between facility i and city j is $c_{ij} + \lambda_i \cdot \frac{f_i}{u_i}$ and the opening cost of facility i is $(1 - \lambda_i) f_i$.

Our algorithm for CFLP is as follows:

1. Construct the UFLP instance (10).
2. Scale the facility costs by a factor of γ_c/γ_f in this instance.
3. Solve the resulting instance using the (γ_f, γ_c)-approximation algorithm.
4. Interpret the output as a solution to CFLP, by opening $\lceil (\sum_{j\in\mathcal{C}} x_{ij})/u_i \rceil$ copies of facility i, for every $i \in \mathcal{F}$.

We prove that the approximation ratio of this algorithm is at most $\gamma_f + \gamma_c$. Consider an optimal solution of the program (10) and denote its facility and connection costs by F_{UFLP} and C_{UFLP}, respectively. By the above argument, $F_{UFLP} + C_{UFLP}$ is a lower bound on the optimal solution of the program (9), which we denote by OPT_{CFLP}. Also, let $SOL = (x_{ij}, y_i)$ denote the solution found in the third step of the above algorithm. If we consider SOL as a solution of the UFLP instance constructed in step 1 of the above algorithm, its facility and connection costs will be

$$F_{SOL} = \sum_{i\in\mathcal{F}} (1-\lambda) f_i y_i \qquad C_{SOL} = \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} (c_{ij} + \lambda \cdot \frac{f_i}{u_i}) x_{ij}. \tag{11}$$

Also, the facility and connection costs of the solution found by the above algorithm is give by

$$F = \sum_{i\in\mathcal{F}} \lceil (\sum_{j\in\mathcal{C}} x_{ij})/u_i \rceil f_i \le \sum_{i\in\mathcal{F}} \left(y_i + (\sum_{j\in\mathcal{C}} x_{ij})/u_i \right) f_i \qquad C = \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} c_{ij} x_{ij}, \tag{12}$$

where the inequality follows from the fact that $\sum_{j\in C} x_{ij} > 0$ implies than $y_i = 1$.

Since the algorithm that we use in step 3 of the above algorithm is a (γ_f, γ_c)-approximation, and the facility costs are scaled by a factor of γ_c/γ_f in step 2, we have

$$\frac{\gamma_c}{\gamma_f} F_{SOL} + C_{SOL} \le \gamma_f \cdot \frac{\gamma_c}{\gamma_f} F_{UFLP} + \gamma_c C_{UFLP} = \gamma_c \cdot (F_{UFLP} + C_{UFLP}) \le \gamma_c \cdot OPT_{CFLP}. \tag{13}$$

On the other hand,

$$\begin{aligned} \frac{\gamma_c}{\gamma_f} F_{SOL} + C_{SOL} &= \frac{\gamma_c}{\gamma_f} \sum_{i\in F} \frac{\gamma_f}{\gamma_f + \gamma_c} f_i y_i + \sum_{i\in F} \sum_{j\in C} (c_{ij} + \frac{\gamma_c f_i}{(\gamma_f + \gamma_c) u_i}) x_{ij} \\ &= \gamma_c \gamma_f + \gamma_c \sum_{i\in F} (\frac{\sum_{j\in C} x_{ij}}{u_i} + y_i) f_i + \sum_{i\in F} \sum_{j\in C} c_{ij} x_{ij} \\ &\ge \frac{\gamma_c}{\gamma_f + \gamma_c} F + C \\ &\ge \frac{\gamma_c}{\gamma_f + \gamma_c} \cdot (F + C) \end{aligned} \tag{14}$$

Inequalities (13) and (14) implies the desired inequality $F + C \leq (\gamma_f + \gamma_c) \cdot OPT_{CFLP}$. □

Theorem 5, together with Lemma 2, imply the main result of this section:

Theorem 6 *The metric CFLP with soft capacities can be approximated within a factor of* 2.89 *in time* $O(n^3)$.

5 Concluding remarks

In Section 3, we proved an upper bound of 1.52 on the approximation factor of our algorithm for UFLP. The reader may ask why we choose the pair $(\gamma_f, \gamma_c) = (1.11, 1.78)$. In fact, we have numerically computed many pairs of (γ_f, γ_c) for $k = 100$ using CPLEX. Then, we have selected the pair to minimize the approximation bound. For example, the pair $(\gamma_f, \gamma_c) = (1.00, 2.00)$ would be easy to prove, but it only gives us a bound of 1.57 on the approximation factor of the algorithm. Similarly, the same pair $(\gamma_f, \gamma_c) = (1.11, 1.78)$ minimizes the approximation ratio $\gamma_f + \gamma_c$ for our CFLP algorithm.

We do not know whether the bound of 1.52 that we proved on the approximation factor of our algorithm is tight or not. The important open question is whether or not our algorithm can close the gap with the approximability lower bound of 1.463 [4]. The main ingredients of our analysis are Lemmas 1 (for the analysis of the first phase of our algorithm) and 3 (for the second phase). Lemma 1 is tight, and the estimate proved in Lemma 2 for the value of γ_c is also very close to the correct value of γ_c. We do not know whether the bound proved in Lemma 3 is tight. It might be possible to apply a method similar to the one used in [6] (i.e., deriving a factor-revealing LP and analyzing it) to analyze both phases of our algorithm in one shot. This might give us a tighter bound on the approximation factor of the algorithm.

Jain et al. [6] show that the existence of a (γ_f, γ_c)-approximation algorithm for UFLP with $\gamma_c < 1+2e^{-\gamma_f}$, would imply that $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log \log n)}]$. Thus, since $\gamma_f+1+2e^{-\gamma_f} \geq 2+\ln 2 > 2.693$, one can not hope to get an approximation ratio better than 2.693 for CFLP using Theorem 5.

Acknowledgment. The authors would like to thank Chaitanya Swamy for pointing out a typo in the earlier draft of the paper.

References

[1] 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.

[2] 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.

[3] F.A. Chudak and D. Shmoys. Improved approximation algorithms for the capacitated facility location problem. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 875–876, 1999.

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

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

[6] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. to appear in STOC 2002, 2002.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

A Proof of Lemma 2

In this appendix, we prove Lemma 2. We first state the following lemma which allows us to restrict our attention to large k's. The proof of this Lemma is exactly the same as the proof of Lemma 12 in [6].

Lemma 7 *If z_k denotes the solution to the factor-revealing LP, then for every k, $z_k \leq z_{2k}$.*

Now, it is enough to prove the following.

Lemma 8 *Let $\gamma_f = 1.11$. Then for every sufficiently large k, the solution of the maximization program (1) is at most 1.78.*

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 $i \leq i'$,

$$(i' - i + 1)\alpha_i - f \leq \sum_{j=i}^{i'} d_j - \sum_{j=1}^{i-1} x_{j,i}. \tag{15}$$

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 (15) for every $i \leq p_2 k$ and $i' = l_i$:

$$(l_i - i + 1)\alpha_i - f \leq \sum_{j=i}^{l_i} d_j - \sum_{j=1}^{i-1} x_{j,i}. \tag{16}$$

For every $i = 1, \ldots, k$, we define θ_i as follows. Here p_3 and p_4 are two constants (with $p_1 < p_3 < 1 - p_3 < p_2$ and $p_4 \leq 1 - p_2$) that will be fixed later.

$$\theta_i = \begin{cases} \frac{1}{l_i - i + 1} & \text{if } i \leq p_3 k \\ \frac{1}{(1-p_3)k} & \text{if } p_3 k < i \leq (1 - p_3)k \\ \frac{p_4 k}{(k-i)(k-i+1)} & \text{if } (1 - p_3)k < i \leq p_2 k \\ 0 & \text{if } i > p_2 k \end{cases} \tag{17}$$

By multiplying both sides of inequality (16) by θ_i and adding up this inequality for $i = 1, \ldots, p_1 k$, $i = p_1 k + 1, \ldots, p_3 k$, $i = p_3 k + 1 \ldots, (1 - p_3)k$, and $i = (1 - p_3)k + 1, \ldots, p_2 k$, we get the following inequalities.

$$\sum_{i=1}^{p_1 k} \alpha_i - (\sum_{i=1}^{p_1 k} \theta_i) f \leq \sum_{i=1}^{p_1 k} \sum_{j=i}^{p_2 k} \frac{d_j}{p_2 k - i + 1} - \sum_{i=1}^{p_1 k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{p_2 k - i + 1} \tag{18}$$

$$\sum_{i=p_1 k+1}^{p_3 k} \alpha_i - (\sum_{i=p_1 k+1}^{p_3 k} \theta_i) f \leq \sum_{i=p_1 k+1}^{p_3 k} \sum_{j=i}^{k} \frac{d_j}{k - i + 1} - \sum_{i=p_1 k+1}^{p_3 k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{k - i + 1} \tag{19}$$

$$\sum_{i=p_3 k+1}^{(1-p_3)k} \frac{k - i + 1}{(1 - p_3)k} \alpha_i - (\sum_{i=p_3 k+1}^{(1-p_3)k} \theta_i) f \leq \sum_{i=p_3 k+1}^{(1-p_3)k} \sum_{j=i}^{k} \frac{d_j}{(1 - p_3)k} - \sum_{i=p_3 k+1}^{(1-p_3)k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{(1 - p_3)k} \tag{20}$$

$$\sum_{i=(1-p_3)k+1}^{p_2k} \frac{p_4k}{k-i}\alpha_i - (\sum_{i=(1-p_3)k+1}^{p_2k} \theta_i)f \leq \sum_{i=(1-p_3)k+1}^{p_2k} \sum_{j=i}^{k} \frac{p_4kd_j}{(k-i)(k-i+1)} - \sum_{i=(1-p_3)k+1}^{p_2k} \sum_{j=1}^{i-1} \frac{p_4k\max(r_{j,i}-d_j,0)}{(k-i)(k-i+1)} \tag{21}$$

We define $s_i := \max_{l \geq i}(\alpha_l - d_l)$. Using this definition and the second and third inequalities of the maximization program (1) we obtain

$$\forall i: \; r_{j,i} \geq s_i - d_j, \text{ which further implies } \max(r_{j,i}-d_j,0) \geq \max(s_i - 2d_j, 0) \tag{22}$$

$$\forall i: \; \alpha_i \leq s_i + d_i \tag{23}$$

$$s_1 \geq s_2 \geq \ldots \geq s_k \; (\geq 0) \tag{24}$$

We assume $s_k \geq 0$ here because that, if on contrary $\alpha_k < d_k$, we can always set α_k equal d_k without violating any constraint in the factor-revealing LP (1) and increase z_k.

Inequality (23) and $p_4 \leq 1 - p_2$ imply

$$\begin{aligned}
&\sum_{i=p_3k+1}^{(1-p_3)k} \left(1 - \frac{k-i+1}{(1-p_3)k}\right)\alpha_i + \sum_{i=(1-p_3)k+1}^{p_2k} \left(1 - \frac{p_4k}{k-i}\right)\alpha_i + \sum_{i=p_2k+1}^{k} \alpha_i \\
&\quad \leq \sum_{i=p_3k+1}^{(1-p_3)k} \frac{i-p_3k-1}{(1-p_3)k}(s_i+d_i) + \sum_{i=(1-p_3)k+1}^{p_2k} \left(1 - \frac{p_4k}{k-i}\right)(s_i+d_i) + \sum_{i=p_2k+1}^{k} (s_i+d_i)
\end{aligned} \tag{25}$$

Let $\zeta := \sum_{i=1}^{k} \theta_i$. Thus,

$$\begin{aligned}
\zeta &= \sum_{i=1}^{p_1k} \frac{1}{p_2k-i+1} + \sum_{i=p_1k+1}^{p_3k} \frac{1}{k-i+1} + \sum_{i=p_3k+1}^{(1-p_3)k} \frac{1}{(1-p_3)k} + \sum_{i=(1-p_3)k+1}^{p_2k} \left(\frac{p_4k}{k-i} - \frac{p_4k}{k-i+1}\right) \\
&= \ln\left(\frac{p_2}{p_2-p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} + o(1).
\end{aligned} \tag{26}$$

By adding the inequalities (18), (19), (20), (21), (25) and using (22), (24), and the fact that $\max(x,0) \geq \delta x$ for every $0 \leq \delta \leq 1$, we obtain

$$\begin{aligned}
\sum_{i=1}^{k} \alpha_i - \zeta f \leq &\sum_{i=1}^{p_1k}\sum_{j=i}^{p_2k} \frac{d_j}{p_2k-i+1} - \sum_{i=1}^{p_1k}\sum_{j=1}^{i-1} \frac{s_i - 2d_j}{2(p_2k-i+1)} \\
&+ \sum_{i=p_1k+1}^{p_3k}\sum_{j=i}^{k} \frac{d_j}{k-i+1} - \sum_{i=p_1k+1}^{p_3k}\sum_{j=1}^{i-1} \frac{s_i-2d_j}{k-i+1} \\
&+ \sum_{i=p_3k+1}^{(1-p_3)k}\sum_{j=i}^{k} \frac{d_j}{(1-p_3)k} - \sum_{i=p_3k+1}^{(1-p_3)k}\sum_{j=1}^{i-1} \frac{s_i-2d_j}{(1-p_3)k} \\
&+ \sum_{i=(1-p_3)k+1}^{p_2k}\sum_{j=i}^{k} \frac{p_4kd_j}{(k-i)(k-i+1)} - \sum_{i=(1-p_3)k+1}^{p_2k}\sum_{j=1}^{i-1} \frac{p_4k\max(s_{p_2k+1}-2d_j,0)}{(k-i)(k-i+1)}
\end{aligned}$$

$$+\sum_{i=p_3k+1}^{(1-p_3)k}\frac{i-p_3k-1}{(1-p_3)k}(s_i+d_i)+\sum_{i=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-i}\right)(s_i+d_i)+\sum_{i=p_2k+1}^{k}(s_{p_2k+1}+d_i)$$

$$=\sum_{j=1}^{p_2k}\sum_{i=1}^{\min(j,p_1k)}\frac{d_j}{p_2k-i+1}-\sum_{i=1}^{p_1k}\frac{i-1}{2(p_2k-i+1)}s_i+\sum_{j=1}^{p_1k}\sum_{i=j+1}^{p_1k}\frac{d_j}{p_2k-i+1}$$

$$+\sum_{j=p_1k+1}^{k}\sum_{i=p_1k+1}^{\min(j,p_3k)}\frac{d_j}{k-i+1}-\sum_{i=p_1k+1}^{p_3k}\frac{i-1}{k-i+1}s_i+\sum_{j=1}^{p_3k}\sum_{i=\max(j,p_1k)+1}^{p_3k}\frac{2d_j}{k-i+1}$$

$$+\sum_{j=p_3k+1}^{k}\sum_{i=p_3k+1}^{\min(j,(1-p_3)k)}\frac{d_j}{(1-p_3)k}-\sum_{i=p_3k+1}^{(1-p_3)k}\frac{i-1}{(1-p_3)k}s_i$$

$$+\sum_{j=1}^{(1-p_3)k}\sum_{i=\max(j,p_3k)+1}^{(1-p_3)k}\frac{2d_j}{(1-p_3)k}$$

$$+\sum_{j=(1-p_3)k+1}^{k}\sum_{i=(1-p_3)k+1}^{\min(j,p_2k)}\left(\frac{1}{k-i}-\frac{1}{k-i+1}\right)p_4kd_j$$

$$-\sum_{j=1}^{p_2k}\sum_{i=\max(j,(1-p_3)k)+1}^{p_2k}p_4k\left(\frac{1}{k-i}-\frac{1}{k-i+1}\right)\max(s_{p_2k+1}-2d_j,0)$$

$$+\sum_{i=p_3k+1}^{(1-p_3)k}\frac{i-p_3k-1}{(1-p_3)k}(s_i+d_i)+\sum_{i=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-i}\right)(s_i+d_i)+\sum_{i=p_2k+1}^{k}d_i$$

$$+(1-p_2)ks_{p_2k+1}$$

$$=\sum_{j=1}^{p_2k}(\mathrm{H}_{p_2k}-\mathrm{H}_{p_2k-\min(j,p_1k)})d_j-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j+\sum_{j=1}^{p_1k}(\mathrm{H}_{p_2k-j}-\mathrm{H}_{(p_2-p_1)k})d_j$$

$$+\sum_{j=p_1k+1}^{k}(\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{k-\min(j,p_3k)})d_j$$

$$-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j+\sum_{j=1}^{p_3k}2(\mathrm{H}_{k-\max(j,p_1k)}-\mathrm{H}_{(1-p_3)k})d_j$$

$$+\sum_{j=p_3k+1}^{k}\frac{\min(j,(1-p_3)k)-p_3k}{(1-p_3)k}d_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{j-1}{(1-p_3)k}s_j$$

$$+\sum_{j=1}^{(1-p_3)k}\frac{2((1-p_3)k-\max(j,p_3k))}{(1-p_3)k}d_j$$

$$+\sum_{j=(1-p_3)k+1}^{k}\left(\frac{1}{k-\min(j,p_2k)}-\frac{1}{p_3k}\right)p_4kd_j$$

$$-\sum_{j=1}^{p_2k}\left(\frac{p_4}{1-p_2}-\frac{p_4k}{k-\max(j,(1-p_3)k)}\right)\max(s_{p_2k+1}-2d_j,0)$$

$$
\begin{aligned}
&+\sum_{j=p_3k+1}^{(1-p_3)k}\frac{j-p_3k-1}{(1-p_3)k}(s_j+d_j)+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)(s_j+d_j)+\sum_{j=p_2k+1}^{k}d_j\\
&+(1-p_2)ks_{p_2k+1}\\
\le&\sum_{j=1}^{p_1k}\left(\mathrm{H}_{p_2k}-\mathrm{H}_{p_2k-j}+\mathrm{H}_{p_2k-j}-\mathrm{H}_{(p_2-p_1)k}+2\mathrm{H}_{(1-p_1)k}-2\mathrm{H}_{(1-p_3)k}+\frac{2(1-2p_3)}{1-p_3}\right)d_j\\
&+\sum_{j=p_1k+1}^{p_3k}\left(\mathrm{H}_{p_2k}-\mathrm{H}_{(p_2-p_1)k}+\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{k-j}+2\mathrm{H}_{k-j}-2\mathrm{H}_{(1-p_3)k}+\frac{2(1-2p_3)}{1-p_3}\right)d_j\\
&+\sum_{j=p_3k+1}^{(1-p_3)k}\left(\mathrm{H}_{p_2k}-\mathrm{H}_{(p_2-p_1)k}+\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{(1-p_3)k}+\frac{j-p_3k}{(1-p_3)k}\right.\\
&\qquad\left.+\frac{2((1-p_3)k-j)}{(1-p_3)k}+\frac{j-p_3k-1}{(1-p_3)k}\right)d_j\\
&+\sum_{j=(1-p_3)k+1}^{p_2k}\left(\mathrm{H}_{p_2k}-\mathrm{H}_{(p_2-p_1)k}+\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{(1-p_3)k}+\frac{1-2p_3}{1-p_3}\right.\\
&\qquad\left.+\frac{p_4k}{k-j}-\frac{p_4k}{p_3k}+\frac{(1-p_4)k-j}{k-j}\right)d_j\\
&+\sum_{j=p_2k+1}^{k}\left(\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{(1-p_3)k}+\frac{1-2p_3}{1-p_3}+\frac{p_4k}{(1-p_2)k}-\frac{p_4k}{p_3k}+1\right)d_j\\
&-\sum_{j=1}^{p_3k}\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\max(s_{p_2k+1}-2d_j,0)-\sum_{j=p_3k+1}^{(1-p_3)k}\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)(s_{p_2k+1}-2d_j)\\
&-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{p_3k}{(1-p_3)k}s_j\\
&+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)s_j+(1-p_2)ks_{p_2k+1} \qquad (27)
\end{aligned}
$$

Let's denote the coefficients of d_j in the above expression by λ_j. Therefore, we have

$$
\begin{aligned}
\sum_{i=1}^{k}\alpha_i-\zeta f\le&\sum_{j=1}^{k}\lambda_jd_j-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{p_3k}{(1-p_3)k}s_j\\
&+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)s_j+\left(1-p_2-(1-2p_3)\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\right)ks_{p_2k+1}\\
&-\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\sum_{j=1}^{p_3k}\max(s_{p_2k+1}-2d_j,0), \qquad (28)
\end{aligned}
$$

where

$$
\lambda_j := \begin{cases}
\ln(\frac{p_2}{p_2-p_1}) + 2\ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1) & \text{if } 1 \le j \le p_1 k \\
\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \mathrm{H}_{k-j} - \mathrm{H}_{(1-p_3)k} + o(1) & \text{if } p_1 k < j \le p_3 k \\
\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} + o(1) & \text{if } p_3 k < j \le (1-p_3)k \\
\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 - \frac{p_4}{p_3} + o(1) & \text{if } (1-p_3)k < j \le p_2 k \\
\ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} + o(1) & \text{if } p_2 k < j \le k.
\end{cases}
$$

For every $j \le p_3 k$, we have

$$
\lambda_{(1-p_3)k} - \lambda_j \le \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} \Rightarrow \delta_j := (\lambda_{(1-p_3)k} - \lambda_j) \Big/ \left(\frac{2p_4}{1-p_2} - \frac{2p_4}{p_3}\right) \le 1. \tag{29}
$$

Also, if we choose p_1, p_2, p_3, p_4 in a way that

$$
\ln(\frac{1-p_1}{1-p_3}) < \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3}, \tag{30}
$$

then for every $j \le p_3 k$, $\lambda_j \le \lambda_{(1-p_3)k}$ and therefore $\delta_j \ge 0$. Then, since $0 \le \delta_j \le 1$, we can replace $\max(s_{p_2k+1} - 2d_j, 0)$ by $\delta_j(s_{p_2k+1} - 2d_j)$ in (28). This gives us

$$
\begin{aligned}
\sum_{i=1}^{k} \alpha_i - \zeta f \le & \sum_{j=1}^{k} \lambda_j d_j - \sum_{j=1}^{p_1 k} \frac{j-1}{2(p_2 k - j + 1)} s_j - \sum_{j=p_1 k+1}^{p_3 k} \frac{j-1}{k-j+1} s_j - \sum_{j=p_3 k+1}^{(1-p_3)k} \frac{p_3 k}{(1-p_3)k} s_j \\
& + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-j}\right) s_j + \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k s_{p_2 k+1} \\
& - \frac{1}{2} \sum_{j=1}^{p_3 k} (\lambda_{(1-p_3)k} - \lambda_j)(s_{p_2 k+1} - 2d_j)
\end{aligned} \tag{31}
$$

Let μ_j denote the coefficient of s_j in the above expression. Therefore the above inequality can be written as

$$
\sum_{i=1}^{k} \alpha_i - \zeta f \le \lambda_{(1-p_3)k} \sum_{j=1}^{(1-p_3)k} d_j + \sum_{j=(1-p_3)k+1}^{k} \lambda_j d_j + \sum_{j=1}^{p_2 k+1} \mu_j s_j, \tag{32}
$$

where

$$
\mu_j = \begin{cases}
-\frac{j-1}{2(p_2 k - j + 1)} & \text{if } 1 \le j \le p_1 k \\
-\frac{j-1}{k-j+1} & \text{if } p_1 k < j \le p_3 k \\
-\frac{p_3}{1-p_3} & \text{if } p_3 k < j \le (1-p_3)k \\
1 - \frac{p_4 k}{k-j} & \text{if } (1-p_3)k < j \le p_2 k
\end{cases} \tag{33}
$$

and

$$\mu_{p_2k+1} = \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right)k - \frac{1}{2}\lambda_{(1-p_3)k}p_3k + \frac{1}{2}\sum_{j=1}^{p_3k}\lambda_j$$
$$= \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right)k - \frac{1}{2}\lambda_{(1-p_3)k}p_3k$$
$$+\frac{p_1k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + 2\ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right)$$
$$+\frac{(p_3-p_1)k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right) + \frac{1}{2}\sum_{j=p_1k+1}^{p_3k}\sum_{i=(1-p_3)k+1}^{k-j}\frac{1}{i}$$
$$= \left(\ln(\frac{1-p_1}{1-p_3}) + 2 - 2p_2 - p_3 + p_1 - 2(1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right) + o(1)\right)\frac{k}{2} \tag{34}$$

Now, if we pick p_1, p_2, p_3, p_4 in such a way that $\lambda_j \leq \gamma$ for every $j \geq (1-p_3)k$, i.e.,

$$\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} < \gamma \tag{35}$$

$$\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 - \frac{p_4}{p_3} < \gamma \tag{36}$$

and

$$\ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < \gamma. \tag{37}$$

then the term $\lambda_{(1-p_3)k}\sum_{j=1}^{(1-p_3)k} d_j + \sum_{j=(1-p_3)k+1}^{k}\lambda_j d_j$ on the right-hand side of (32) is at most $\gamma\sum_{j=1}^{k} d_j$. Also, if for every $i \leq p_2k+1$, we have

$$\mu_1 + \mu_2 + \cdots + \mu_i \leq 0, \tag{38}$$

then by inequality (24), we have $\sum_{j=1}^{p_2k+1}\mu_j s_j \leq 0$. Therefore, if p_1, p_2, p_3, p_4 are chosen in such a way that in addition to the above inequalities, we have

$$\ln\left(\frac{p_2}{p_2-p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < 1.11, \tag{39}$$

then inequality (32) can be written as

$$\sum_{i=1}^{k}\alpha_i - 1.11f \leq \gamma\sum_{j=1}^{k} d_j, \tag{40}$$

which shows that the solution of the maximization program (1) is at most γ. From (33), it is clear that $\mu_j \leq 0$ for every $j \leq (1-p_3)k$ and $\mu_j \geq 0$ for every $(1-p_3)k \leq j \leq p_2k$. Therefore, it is enough to check inequality (38) for $i = p_2k$ and $i = p_2k+1$. We have

$$
\begin{aligned}
\sum_{j=1}^{p_2k}\mu_j =& -\sum_{j=1}^{p_1k}\frac{p_2k-p_2k+j-1}{2(p_2k-j+1)} - \sum_{j=p_1k+1}^{p_3k}\frac{k-k+j-1}{k-j+1} - \frac{p_3(1-2p_3)k}{1-p_3} \\
& +(p_2-1+p_3)k - \sum_{j=(1-p_3)k+1}^{p_2k}\frac{p_4k}{k-j} \\
=& -\frac{p_2k}{2}(\mathrm{H}_{p_2k}-\mathrm{H}_{(p_2-p_1)k}) + \frac{p_1k}{2} - k(\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{(1-p_3)k}) + (p_3-p_1)k \\
& -\frac{p_3(1-2p_3)k}{1-p_3} + (p_2-1+p_3)k - p_4k(\mathrm{H}_{p_3k}-\mathrm{H}_{(1-p_2)k}) \\
=& \Big(-\frac{p_1}{2}+p_2+2p_3-1-\frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} \\
& -p_4\ln(\frac{p_3}{1-p_2}) + o(1)\Big)\, k \qquad (41)
\end{aligned}
$$

Therefore, inequality (38) is equivalent to the following two inequalities.

$$
-\frac{p_1}{2}+p_2+2p_3-1-\frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4\ln(\frac{p_3}{1-p_2}) < 0 \qquad (42)
$$

$$
\begin{aligned}
& -\frac{p_1}{2}+p_2+2p_3-1-\frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4\ln(\frac{p_3}{1-p_2}) \\
& \quad +\frac{1}{2}\ln(\frac{1-p_1}{1-p_3}) + 1 - p_2 - \frac{p_3}{2} + \frac{p_1}{2} - (1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right) < 0 \qquad (43)
\end{aligned}
$$

Now, it is enough to observe that if we let $p_1 = 0.225, p_2 = 0.791, p_3 = 0.305$, $p_4 = 0.06984$, and $\gamma = 1.7764$, then inequalities (30), (35), (36), (37), (39), (42), and (43) are all satisfied. Therefore, the solution of the optimization program (1) is at most $1.7764 < 1.78$. □

Remark: Numerical computations using CPLEX show that $z_{500} \approx 1.7743$ and therefore $\gamma_c > 1.774$ for $\gamma_f = 1.11$. Thus, the estimate provided by Lemma 2 for the value of γ_c is close to its actual value.