A 1.52-Approximation Algorithm for the Uncapacitated Facility Location Problem

Mohammad Mahdian* Yinyu Ye† Jiawei Zhang†

January 2, 2002

Abstract

In this note we present an improved approximation algorithm for the (uncapacitated) metric facility location problem. This algorithm uses the idea of cost scaling, the greedy algorithm of [5], and the greedy augmentation procedure of [1, 3].

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 [3] proved that it is impossible to get an approximation guarantee of 1.463 for the metric facility location problem, unless $\mathbf{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. For a more detailed survey on this problem, see [9].

In this note, we combine the greedy algorithm of [5] and the greedy augmentation of [1, 3] to show that UFLP can be approximated within a factor of 1.52, whereas the previously known best factor was 1.582 [11]. Note that this approximation factor is very close to the lower bound of 1.463 proved in [3].

The algorithm 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. Some possible directions for the future research on this problem are discussed in Section 4.

*Department of Mathematics, MIT, Cambridge, MA 02139, USA. E-mail: mahdian@math.mit.edu.

†Department of Management Sciences, Henry B. Tippie College of Business, The University of Iowa, Iowa City, IA 52242, USA. E-mail: {yinyu-ye,jiawei-zhang}@uiowa.edu. Research supported in part by NSF grants DMI-9908077.

approx. factor	reference	technique/running time
$O(\ln n_c)$	Hochbaum [4]	greedy algorithm/$O(n^3)$
3.16	Shmoys et al. [10]	LP rounding
2.47	Guha and Khuller [3]	LP rounding + greedy augmentation
1.736	Chudak [2]	LP rounding
$5+\epsilon$	Korupolu et al. [7]	local search/$O(n^6 \log(n/\epsilon))$
3	Jain and Vazirani [6]	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. [8]	greedy algorithm/$O(n^2 \log n)$
1.61	Jain et al. [5]	greedy algorithm/$O(n^3)$
1.582	Sviridenko [11]	LP rounding
1.52	This paper	greedy algorithm + greedy augmentation/$O(n^3)$

Table 1: Approximation Algorithms for UFLP

2 Algorithm

In [5], Jain, Mahdian, and Saberi proposed a greedy algorithm for the facility location problem 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 closed, and the *budget* of every city j, denoted by B_j, is initialized to 0. At every moment, each city j offers some money from its budget to each *closed* facility i. The amount of this offer is 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 closed facility i, the total offer that it receives from cities is equal to the cost of opening i. In this case, we open facility i, and for every city j (connected or unconnected) which has a non-zero offer to i, we connect j to i.

   (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 closed 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. [8].

Here we use the JMS algorithm to solve the facility location problem with an improved approximation factor. In the *first* stage of our algorithm, we scale up the facility costs 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 in [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 stage of the algorithm.

In the *second* stage of the algorithm, we scale down the facility costs back to their original values. 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 or less than 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 [3] 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 [5] and [1] that bound the cost of the solution found by the JMS algorithm and the greedy augmentation procedure. The following theorem gives tight bounds on the cost of the solution found by the JMS algorithm in terms of the facility and connection costs of an arbitrary solution.

Lemma 1 ([5]) *Let $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program.*

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

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 the JMS algorithm is at most $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.

In particular, it is proved in [5] that $\gamma_c \leq 1.61$ when $\gamma_f = 1.61$, and therefore the JMS algorithm is a 1.61-approximation algorithm. Here we use the above theorem with $\gamma_f = 1.11$. We prove the following lemma for this case. 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, $\gamma_c \leq 1.78$ when $\gamma_f = 1.11$.*

We also use the following result of [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 the facility location problem 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 ($\delta \geq 1$). 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

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

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

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

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

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

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,

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

Inequalities (5) and (8) show that in each 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 facility location problem. It is easy to see that this algorithm can be implemented in $O(n^3)$ time. $\square$

4 Concluding remarks

In the previous section, we proved an upper bound of 1.52 on the approximation factor of our algorithm. 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 $k = 100$ by linear programming. Then, we choose 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 possesses the bound 1.57.

We don't know whether this bound 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 proved in [3]. The main ingredients of our analysis are Lemmas 1 and 3. Lemma 1 is tight, and the estimate proved in Lemma 2 for the value of γ_c is also close to the correct value of γ_c. We don't know whether the bound proved in Lemma 3 is tight. Also, it might be possible to apply a method similar to the one used in [5] for the analysis of the combined algorithm (i.e., deriving a factor-LP and analyzing it) to obtain a tighter bound on the approximation factor of our algorithm.

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

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

[5] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. manuscript, 2001.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] M. Sviridenko. An 1.582-approximation algorithm for the metric uncapacitated facility location problem. manuscript, 2001.

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 14 in [5].

Lemma 5 *If z_k denotes the solution to the factor LP, then for every k, $z_k \leq z_{2k}$.*

Now, it is enough to prove the following.

Lemma 6 *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 LP. Let $x_{j,i} := max(r_{j,i} - d_j, 0)$. The fourth inequality of the factor LP implies that for every $i \le i'$,

$$(i' - i + 1)\alpha_i - f \le \sum_{j=i}^{i'} d_j - \sum_{j=1}^{i-1} x_{j,i}. \tag{9}$$

Now, we define l_i as follows:

$$l_i = \begin{cases} p_2k & \text{if } i \le p_1k \\ k & \text{if } i > p_1k \end{cases}$$

where p_1 and p_2 are two constants (with $p_1 < p_2$) that will be fixed later. Consider Inequality (9) for every $i \le p_2k$ and $i' = l_i$:

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

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 $p_4 \le 1 - p_2$) that will be fixed later.

$$\theta_i = \begin{cases} \frac{1}{l_i - i + 1} & \text{if } i \le p_3k \\ \frac{1}{(1-p_3)k} & \text{if } p_3k < i \le (1-p_3)k \\ \frac{p_4k}{(k-i)(k-i+1)} & \text{if } (1-p_3)k < i \le p_2k \\ 0 & \text{if } i > p_2k \end{cases} \tag{11}$$

By multiplying both sides of inequality (10) by θ_i and adding up this inequality for $i = 1, \ldots, p_1k$, $i = p_1k + 1, \ldots, p_3k$, $i = p_3k + 1 \ldots, (1 - p_3)k$, and $i = (1 - p_3)k + 1, \ldots, p_2k$, we get the following inequalities.

$$\sum_{i=1}^{p_1k} \alpha_i - (\sum_{i=1}^{p_1k} \theta_i)f \le \sum_{i=1}^{p_1k}\sum_{j=i}^{p_2k} \frac{d_j}{p_2k - i + 1} - \sum_{i=1}^{p_1k}\sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{p_2k - i + 1} \tag{12}$$

$$\sum_{i=p_1k+1}^{p_3k} \alpha_i - (\sum_{i=p_1k+1}^{p_3k} \theta_i)f \le \sum_{i=p_1k+1}^{p_3k}\sum_{j=i}^{k} \frac{d_j}{k - i + 1} - \sum_{i=p_1k+1}^{p_3k}\sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{k - i + 1} \tag{13}$$

$$\sum_{i=p_3k+1}^{(1-p_3)k} \frac{k - i + 1}{(1-p_3)k}\alpha_i - (\sum_{i=p_3k+1}^{(1-p_3)k} \theta_i)f \le \sum_{i=p_3k+1}^{(1-p_3)k}\sum_{j=i}^{k} \frac{d_j}{(1-p_3)k} - \sum_{i=p_3k+1}^{(1-p_3)k}\sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{(1-p_3)k} \tag{14}$$

$$\begin{aligned}\sum_{i=(1-p_3)k+1}^{p_2k} \frac{p_4k}{k-i}\alpha_i - (\sum_{i=(1-p_3)k+1}^{p_2k} \theta_i)f \le &\sum_{i=(1-p_3)k+1}^{p_2k}\sum_{j=i}^{k} \frac{p_4kd_j}{(k-i)(k-i+1)} \\ &- \sum_{i=(1-p_3)k+1}^{p_2k}\sum_{j=1}^{i-1} \frac{p_4k\max(r_{j,i} - d_j, 0)}{(k-i)(k-i+1)}\end{aligned} \tag{15}$$

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

$$\forall i: \ r_{j,i} \geq s_i - d_j, \text{ which further implies } \max(r_{j,i} - d_j, 0) \geq \max(s_i - 2d_j, 0) \tag{16}$$

$$s_1 \geq s_2 \geq \ldots \geq s_k \ (\geq 0) \tag{17}$$

$$\forall i: \ \alpha_i \leq s_i + d_i \tag{18}$$

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 LP (1) and increase z_k.

Inequality (18) and $p_4 \leq 1 - p_2$ imply

$$\begin{aligned}
&\sum_{i=p_3k+1}^{(1-p_3)k} \left(1 - \frac{k-i+1}{(1-p_3)k}\right) \alpha_i + \sum_{i=(1-p_3)k+1}^{p_2k} \left(1 - \frac{p_4k}{k-i}\right) \alpha_i + \sum_{i=p_2k+1}^{k} \alpha_i \\
&\quad \leq \sum_{i=p_3k+1}^{(1-p_3)k} \frac{i - p_3k - 1}{(1-p_3)k}(s_i + d_i) + \sum_{i=(1-p_3)k+1}^{p_2k} \left(1 - \frac{p_4k}{k-i}\right)(s_i + d_i) + \sum_{i=p_2k+1}^{k} (s_i + d_i)
\end{aligned} \tag{19}$$

Let

$$\begin{aligned}
\zeta &:= \sum_{i=1}^{k} \theta_i \\
&= \sum_{i=1}^{p_1k} \frac{1}{p_2k - i + 1} + \sum_{i=p_1k+1}^{p_3k} \frac{1}{k-i+1} + \sum_{i=p_3k+1}^{(1-p_3)k} \frac{1}{(1-p_3)k} + \sum_{i=(1-p_3)k+1}^{p_2k} \left(\frac{p_4k}{k-i} - \frac{p_4k}{k-i+1}\right) \\
&= \ln\left(\frac{p_2}{p_2 - p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} + o(1).
\end{aligned} \tag{20}$$

By adding the inequalities (12), (13), (14), (15), (19) and using (16), (17), and the fact that $\max(x, 0) \geq \delta x$ for every $0 \leq \delta \leq 1$, we obtain

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

$$
\begin{aligned}
=&\sum_{j=1}^{p_2k}\sum_{i=1}^{\min(j,p_1k)}\frac{d_j}{p_2k-i+1}-\sum_{i=1}^{p_1k}\frac{i-1}{2(p_2k-i+1)}s_i+\sum_{j=1}^{p_1k}\sum_{i=j+1}^{p_1k}\frac{d_j}{p_2k-i+1}\\
&+\sum_{j=p_1k+1}^{k}\sum_{i=p_1k+1}^{\min(j,p_3k)}\frac{d_j}{k-i+1}-\sum_{i=p_1k+1}^{p_3k}\frac{i-1}{k-i+1}s_i+\sum_{j=1}^{p_3k}\sum_{i=\max(j,p_1k)+1}^{p_3k}\frac{2d_j}{k-i+1}\\
&+\sum_{j=p_3k+1}^{k}\sum_{i=p_3k+1}^{\min(j,(1-p_3)k)}\frac{d_j}{(1-p_3)k}-\sum_{i=p_3k+1}^{(1-p_3)k}\frac{i-1}{(1-p_3)k}s_i\\
&+\sum_{j=1}^{(1-p_3)k}\sum_{i=\max(j,p_3k)+1}^{(1-p_3)k}\frac{2d_j}{(1-p_3)k}\\
&+\sum_{j=(1-p_3)k+1}^{k}\sum_{i=(1-p_3)k+1}^{\min(j,p_2k)}\left(\frac{1}{k-i}-\frac{1}{k-i+1}\right)p_4kd_j\\
&-\sum_{j=1}^{p_2k}\sum_{i=\max(j,(1-p_3)k)+1}^{p_2k}p_4k\left(\frac{1}{k-i}-\frac{1}{k-i+1}\right)\max(s_{p_2k+1}-2d_j,0)\\
&+\sum_{i=p_3k+1}^{(1-p_3)k}\frac{i-p_3k-1}{(1-p_3)k}(s_i+d_i)+\sum_{i=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-i}\right)(s_i+d_i)+\sum_{i=p_2k+1}^{k}d_i\\
&+(1-p_2)ks_{p_2k+1}\\
=&\sum_{j=1}^{p_2k}(\mathrm{H}_{p_2k}-\mathrm{H}_{p_2k-\min(j,p_1k)})d_j-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j+\sum_{j=1}^{p_1k}(\mathrm{H}_{p_2k-j}-\mathrm{H}_{(p_2-p_1)k})d_j\\
&+\sum_{j=p_1k+1}^{k}(\mathrm{H}_{(1-p_1)k}-\mathrm{H}_{k-\min(j,p_3k)})d_j\\
&-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j+\sum_{j=1}^{p_3k}2(\mathrm{H}_{k-\max(j,p_1k)}-\mathrm{H}_{(1-p_3)k})d_j\\
&+\sum_{j=p_3k+1}^{k}\frac{\min(j,(1-p_3)k)-p_3k}{(1-p_3)k}d_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{j-1}{(1-p_3)k}s_j\\
&+\sum_{j=1}^{(1-p_3)k}\frac{2((1-p_3)k-\max(j,p_3k))}{(1-p_3)k}d_j\\
&+\sum_{j=(1-p_3)k+1}^{k}\left(\frac{1}{k-\min(j,p_2k)}-\frac{1}{p_3k}\right)p_4kd_j\\
&-\sum_{j=1}^{p_2k}\left(\frac{p_4}{1-p_2}-\frac{p_4k}{k-\max(j,(1-p_3)k)}\right)\max(s_{p_2k+1}-2d_j,0)\\
&+\sum_{j=p_3k+1}^{(1-p_3)k}\frac{j-p_3k-1}{(1-p_3)k}(s_j+d_j)+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)(s_j+d_j)+\sum_{j=p_2k+1}^{k}d_j\\
&+(1-p_2)ks_{p_2k+1}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{j=1}^{p_1 k} \left(\mathrm{H}_{p_2 k} - \mathrm{H}_{p_2 k - j} + \mathrm{H}_{p_2 k - j} - \mathrm{H}_{(p_2 - p_1) k} + 2\mathrm{H}_{(1-p_1)k} - 2\mathrm{H}_{(1-p_3)k} + \frac{2(1-2p_3)}{1-p_3} \right) d_j \\
&\quad + \sum_{j=p_1 k+1}^{p_3 k} \left(\mathrm{H}_{p_2 k} - \mathrm{H}_{(p_2 - p_1) k} + \mathrm{H}_{(1-p_1)k} - \mathrm{H}_{k-j} + 2\mathrm{H}_{k-j} - 2\mathrm{H}_{(1-p_3)k} + \frac{2(1-2p_3)}{1-p_3} \right) d_j \\
&\quad + \sum_{j=p_3 k+1}^{(1-p_3) k} \left(\mathrm{H}_{p_2 k} - \mathrm{H}_{(p_2 - p_1) k} + \mathrm{H}_{(1-p_1)k} - \mathrm{H}_{(1-p_3)k} + \frac{j - p_3 k}{(1-p_3)k} \right. \\
&\qquad\qquad \left. + \frac{2((1-p_3)k - j)}{(1-p_3)k} + \frac{j - p_3 k - 1}{(1-p_3)k} \right) d_j \\
&\quad + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(\mathrm{H}_{p_2 k} - \mathrm{H}_{(p_2 - p_1) k} + \mathrm{H}_{(1-p_1)k} - \mathrm{H}_{(1-p_3)k} + \frac{1-2p_3}{1-p_3} \right. \\
&\qquad\qquad \left. + \frac{p_4 k}{k-j} - \frac{p_4 k}{p_3 k} + \frac{(1-p_4)k - j}{k-j} \right) d_j \\
&\quad + \sum_{j=p_2 k+1}^{k} \left(\mathrm{H}_{(1-p_1)k} - \mathrm{H}_{(1-p_3)k} + \frac{1-2p_3}{1-p_3} + \frac{p_4 k}{(1-p_2)k} - \frac{p_4 k}{p_3 k} + 1 \right) d_j \\
&\quad - \sum_{j=1}^{p_3 k} \left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3} \right) \max(s_{p_2 k+1} - 2d_j, 0) - \sum_{j=p_3 k+1}^{(1-p_3)k} \left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3} \right) (s_{p_2 k+1} - 2d_j) \\
&\quad - \sum_{j=1}^{p_1 k} \frac{j-1}{2(p_2 k - j + 1)} s_j - \sum_{j=p_1 k+1}^{p_3 k} \frac{j-1}{k-j+1} s_j - \sum_{j=p_3 k+1}^{(1-p_3)k} \frac{p_3 k}{(1-p_3)k} s_j \\
&\quad + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-j} \right) s_j + (1-p_2) k s_{p_2 k+1} \qquad (21)
\end{aligned}
$$

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

$$
\begin{aligned}
\sum_{i=1}^{k} \alpha_i - \zeta f &\leq \sum_{j=1}^{k} \lambda_j d_j - \sum_{j=1}^{p_1 k} \frac{j-1}{2(p_2 k - j + 1)} s_j - \sum_{j=p_1 k+1}^{p_3 k} \frac{j-1}{k-j+1} s_j - \sum_{j=p_3 k+1}^{(1-p_3)k} \frac{p_3 k}{(1-p_3)k} s_j \\
&\quad + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-j} \right) s_j + \left(1 - p_2 - (1-2p_3) \left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3} \right) \right) k s_{p_2 k+1} \\
&\quad - \left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3} \right) \sum_{j=1}^{p_3 k} \max(s_{p_2 k+1} - 2d_j, 0), \qquad (22)
\end{aligned}
$$

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

$$\lambda_{(1-p_3)k} - \lambda_j \le \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} \Rightarrow \delta_j := (\lambda_{(1-p_3)k} - \lambda_j) \Big/ \left(\frac{2p_4}{1-p_2} - \frac{2p_4}{p_3}\right) \le 1. \tag{23}$$

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

$$\ln(\frac{1-p_1}{1-p_3}) < \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3}, \tag{24}$$

then for every $j \le p_3 k$, $\lambda_j \le \lambda_{(1-p_3)k}$ and therefore $\delta_j \ge 0$. Thus, 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 (22). This gives us

$$\begin{aligned} \sum_{i=1}^{k} \alpha_i - \zeta f \le & \sum_{j=1}^{k} \lambda_j d_j - \sum_{j=1}^{p_1 k} \frac{j-1}{2(p_2 k - j + 1)} s_j - \sum_{j=p_1k+1}^{p_3 k} \frac{j-1}{k-j+1} s_j - \sum_{j=p_3k+1}^{(1-p_3)k} \frac{p_3 k}{(1-p_3)k} s_j \\ & + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-j}\right) s_j + \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k s_{p_2k+1} \\ & - \frac{1}{2} \sum_{j=1}^{p_3 k} (\lambda_{(1-p_3)k} - \lambda_j)(s_{p_2k+1} - 2d_j) \end{aligned} \tag{25}$$

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

$$\sum_{i=1}^{k} \alpha_i - \zeta f \le \lambda_{(1-p_3)k} \sum_{j=1}^{(1-p_3)k} d_j + \sum_{j=(1-p_3)k+1}^{k} \lambda_j d_j + \sum_{j=1}^{p_2k+1} \mu_j s_j \tag{26}$$

We can compute the value of μ_{p_2k+1} as follows.

$$\begin{aligned} \mu_{p_2k+1} &= \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k - \frac{1}{2}\lambda_{(1-p_3)k} p_3 k + \frac{1}{2} \sum_{j=1}^{p_3 k} \lambda_j \\ &= \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k - \frac{1}{2}\lambda_{(1-p_3)k} p_3 k \end{aligned}$$

$$
\begin{aligned}
&+\frac{p_1 k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + 2\ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right) \\
&+\frac{(p_3-p_1)k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right) + \frac{1}{2}\sum_{j=p_1k+1}^{p_3k}\sum_{i=(1-p_3)k+1}^{k-j}\frac{1}{i} \\
=& \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right)k \\
&-\frac{p_3 k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} + o(1)\right) \\
&+\frac{p_3 k}{2}\ln(\frac{p_2}{p_2-p_1}) + \frac{(p_3+p_1)k}{2}\ln(\frac{1-p_1}{1-p_3}) \\
&+\frac{p_3(1-2p_3)k}{1-p_3} + \frac{1}{2}\sum_{i=(1-p_3)k+1}^{(1-p_1)k}\frac{k-i-p_1k}{i} + o(1) \\
=& \frac{k}{2}\ln(\frac{1-p_1}{1-p_3}) + (1 - p_2 - \frac{p_3}{2} + \frac{p_1}{2})k - (1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)k + o(1)
\end{aligned} \tag{27}
$$

Thus, we can summarize the values of μ_j's in the following.

$$
\mu_j = \begin{cases}
-\dfrac{j-1}{2(p_2k-j+1)} & \text{if } 1 \le j \le p_1 k \\
-\dfrac{j-1}{k-j+1} & \text{if } p_1 k < j \le p_3 k \\
-\dfrac{p_3}{1-p_3} & \text{if } p_3 k < j \le (1-p_3)k \\
1 - \dfrac{p_4 k}{k-j} & \text{if } (1-p_3)k < j \le p_2 k \\
\left(\ln(\dfrac{1-p_1}{1-p_3}) + 2 - 2p_2 - p_3 + p_1 - 2(1-p_3)\left(\dfrac{p_4}{1-p_2} - \dfrac{p_4}{p_3}\right) + o(1)\right)\dfrac{k}{2} & \text{if } j = p_2 k + 1
\end{cases} \tag{28}
$$

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

$$
\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} < \gamma \tag{29}
$$

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

and

$$
\ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < \gamma. \tag{31}
$$

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 (26) is at most $\gamma\sum_{j=1}^{k} d_j$. Also, if for every $i \le p_2 k + 1$, we have

$$
\mu_1 + \mu_2 + \cdots + \mu_i \le 0, \tag{32}
$$

then by inequality (17), we have $\sum_{j=1}^{p_2k+1} \mu_j s_j \le 0$. 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

$$\ln\left(\frac{p_2}{p_2-p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < 1.11, \tag{33}$$

then inequality (26) can be written as

$$\sum_{i=1}^{k} \alpha_i - 1.11f \le \gamma \sum_{j=1}^{k} d_j, \tag{34}$$

which shows that the solution of the maximization program (1) is at most γ. ¿From (28), it is clear that $\mu_j \le 0$ for every $j \le (1-p_3)k$ and $\mu_j \ge 0$ for every $(1-p_3)k \le j \le p_2k$. Therefore, it is enough to check inequality (32) for $i = p_2k$ and $i = p_2k+1$. We have

$$\begin{aligned}
\sum_{j=1}^{p_2k} \mu_j &= -\sum_{j=1}^{p_1k} \frac{p_2k - p_2k + j - 1}{2(p_2k - j + 1)} - \sum_{j=p_1k+1}^{p_3k} \frac{k-k+j-1}{k-j+1} - \frac{p_3(1-2p_3)k}{1-p_3} \\
&\quad +(p_2 - 1 + p_3)k - \sum_{j=(1-p_3)k+1}^{p_2k} \frac{p_4k}{k-j} \\
&= -\frac{p_2k}{2}(\mathrm{H}_{p_2k} - \mathrm{H}_{(p_2-p_1)k}) + \frac{p_1k}{2} - k(\mathrm{H}_{(1-p_1)k} - \mathrm{H}_{(1-p_3)k}) + (p_3 - p_1)k \\
&\quad -\frac{p_3(1-2p_3)k}{1-p_3} + (p_2 - 1 + p_3)k - p_4k(\mathrm{H}_{p_3k} - \mathrm{H}_{(1-p_2)k}) \\
&= \left(\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3}\right. \\
&\quad \left. -p_4\ln(\frac{p_3}{1-p_2}) + o(1)\right) k
\end{aligned} \tag{35}$$

Therefore, inequality (32) is equivalent to the following two inequalities.

$$\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4\ln(\frac{p_3}{1-p_2}) < 0 \tag{36}$$

$$\begin{aligned}
&\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4\ln(\frac{p_3}{1-p_2}) \\
&\quad +\frac{1}{2}\ln(\frac{1-p_1}{1-p_3}) + 1 - p_2 - \frac{p_3}{2} + \frac{p_1}{2} - (1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right) < 0
\end{aligned} \tag{37}$$

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 (24), (29), (30), (31), (33), (36), and (37) are all satisfied. Therefore, the solution of the optimization program (1) is at most $1.7764 < 1.78$. □

Remark: Numerical computations using AMPL/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.