Approximation Algorithms for Metric Facility Location Problems *

Mohammad Mahdian[†] Yinyu Ye[‡] Jiawei Zhang [§]

Abstract

In this paper we present a 1.52-approximation algorithm for the metric uncapacitated facility location problem, and a 2-approximation algorithm for the metric capacitated facility location problem with soft capacities. Both these algorithms improve the best previously known approximation factor for the corresponding problem, and our soft-capacitated facility location algorithm achieves the integrality gap of the standard LP relaxation of the problem. Furthermore, we will show, using a result of Thorup, that our algorithms can be implemented in quasi-linear time.

Keyword: approximation algorithms, facility location problem, greedy method, linear programming

1 Introduction

Variants of the facility location problem (FLP) have been studied extensively in the operations research and management science literatures and have received considerable attention in the area of approximation algorithms (See [21] for a survey). In the metric uncapacitated facility location problem (UFLP), which is the most basic facility location problem, we are given a set $\mathcal{F}$ of *facilities*, a set $\mathcal{C}$ of *cities* (a.k.a. clients), a cost f_i for opening facility $i \in \mathcal{F}$, and a connection cost c_{ij} for connecting client j to 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, that is, the cost of opening facilities and connecting the clients, is minimized. We assume that the connection costs form a metric, meaning that they are symmetric and satisfy the triangle inequality.

Since the first constant factor approximation algorithm due to Shmoys, Tardos and Aardal [22], a large number of approximation algorithms have been proposed for UFLP [23, 12, 25, 14, 2, 4, 6, 8, 14, 15]. Table 1 shows a summary of these results. Prior to this work, the best known approximation factor for UFLP was 1.58, given by Sviridenko [23], which was achieved using LP rounding. Guha and Khuller [8] proved that it is impossible to get an approximation guarantee of 1.463 for UFLP, unless **NP** $\subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. In this paper, we give a 1.52-approximation algorithm for UFLP which can be implemented in quasi-linear time, using a result of Thorup [24]. Our algorithm combines the greedy algorithm of Jain, Mahdian, and Saberi [13, 12] with the idea of cost scaling, and is analyzed using a factor-revealing LP.

*This paper is based on preliminary versions [18, 19].

†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 grant DMI-0231600.

§IOMS-Operations Management, Stern School of Business, New York University, 44 West 4th Street, Suite 8-66, New York, NY 10012-1126. Email: jzhang@stern.nyu.edu. Research supported in part by NSF grant DMI-0231600.

approx. factor	reference	technique/running time
$O(\ln n_c)$	Hochbaum [10]	greedy algorithm/$O(n^3)$
3.16	Shmoys et al. [22]	LP rounding
2.41	Guha and Khuller [8]	LP rounding + greedy augmentation
1.736	Chudak and Shmoys [6]	LP rounding
$5+\epsilon$	Korupolu et al. [15]	local search/$O(n^6 \log(n/\epsilon))$
3	Jain and Vazirani [14]	primal-dual method/$O(n^2 \log n)$
1.853	Charikar and Guha [4]	primal-dual method + greedy augmentation/$O(n^3)$
1.728	Charikar and Guha [4]	LP rounding + primal-dual method + greedy augmentation
1.861	Mahdian et al. [16, 12]	greedy algorithm/$O(n^2 \log n)$
1.61	Jain et al. [13, 12]	greedy algorithm/$O(n^3)$
1.582	Sviridenko [23]	LP rounding
1.52	This paper	greedy algorithm + cost scaling/$\tilde{O}(n)$

Table 1: Approximation Algorithms UFLP

The growing interest in UFLP is not only due to its applications in a large number of settings [7], but also due to the fact that UFLP is one of the most basic models among discrete location problems. The insights gained in dealing with UFLP may also apply to more complicated location models, and in many cases the latter can be reduced directly to UFLP.

In the second part of this paper, we give a 2-approximation algorithm for the soft-capacitated facility location problem (SCFLP) by reducing it to UFLP. SCFLP is similar to UFLP, except that there is a capacity u_i associated with each facility i, which means that if we want this facility to serve x cities, we have to open it $\lceil x/u_i \rceil$ times at a cost of $f_i \lceil x/u_i \rceil$. This problem is also known as the facility location problem with integer decision variables in the operations research literature (See [3] and [20]). Chudak and Shmoys [5] gave a 3-approximation algorithm for SCFLP with uniform capacities (i.e., $u_i = u$ for all $i \in \mathcal{F}$) using LP rounding. For non-uniform capacities, Jain and Vazirani [14] showed how to reduce this problem to UFLP, and by solving UFLP through a primal-dual algorithm, they obtained a 4-approximation. Arya et al [2] proposed a local search algorithm that achieves an approximation ratio of 3.72. Following the approach of Jain and Vazirani [14], Jain, Mahdian, and Saberi [13, 12] showed that SCFLP can be approximated within a factor of 3. This was the best previously known algorithm for this problem. We improve this factor to 2, achieving the integrality gap of the natural LP relaxation of the problem. The main idea of our algorithm is to consider algorithms and reductions that have separate (not necessarily equal) approximation factors for the facility and connection costs. We will define the concept of *bifactor* approximate reduction in this paper, and show how it can be used to get an approximation factor of 2 for SCFLP. The idea of using bifactor approximation algorithms and reductions can be used to improve the approximation factor of several other problems.

The rest of this paper is organized as follows: In Section 2 the necessary definitions and notations are presented. In Section 3, we present the algorithm for UFLP and its underlying intuition, and we prove the upper bound of 1.52 on the approximation factor of the algorithm. In Section 4 we present a lemma on the approximability of the linear-cost facility location problem. In Section 5 we define the concept of a bifactor approximate reduction between facility location problems. Using bifactor reductions to the linear-cost FLP and the lemma proved in Section 4, we present algorithms for SCFLP and the concave soft-capacitated FLP. Concluding remarks are given in Section 6.

2 Preliminaries

In this paper, we will define reductions between various facility location problems. Many such problems can be considered as special cases of the *universal facility location problem*, as defined below. This problem was first defined [9] and further studied in [17].

Definition 1 In the *metric universal facility location problem*, we are given a set $\mathcal{C}$ of n_c cities, a set $\mathcal{F}$ of n_f facilities, a connection cost c_{ij} between city j and facility i for every $i \in \mathcal{F}, j \in \mathcal{C}$, and a facility cost function $f_i : \{0, \ldots, n_c\} \mapsto \mathbb{R}^+$ for every $i \in \mathcal{F}$. Connection costs are symmetric and obey the triangle inequality. The value of $f_i(k)$ equals the cost of opening facility i, if it is used to serve k cities. A solution to the problem is a function $\phi : \mathcal{C} \to \mathcal{F}$ assigning each city to a facility. The *facility cost F_ϕ* of the solution ϕ is defined as $\sum_{i \in \mathcal{F}} f_i(|\{j : \phi(j) = i\}|)$, i.e., the total cost for opening facilities. The *connection cost* (a.k.a. service cost) C_ϕ of ϕ is $\sum_{j \in \mathcal{C}} c_{\phi(j),j}$, i.e., the total cost of opening each city to its assigned facility. The objective is to find a solution ϕ that minimizes the sum $F_\phi + C_\phi$.

For the metric universal facility location problem, we distinguish two models by how the connection costs are given. In the distance oracle model, the connection costs are explicitly given by a matrix (c_{ij}) for any $i \in \mathcal{F}$ and $j \in \mathcal{C}$. In the sparse graph model, $\mathcal{C}$ and $\mathcal{F}$ are nodes of an undirected graph (which may not be complete) in which the cost of each edge is given, and the connection cost between a facility i and a client j is implicitly given by the shortest distance between i and j.

Now we can define the uncapacitated and soft-capacitated facility location problems as special cases of the universal FLP:

Definition 2 *The* metric uncapacitated facility location problem (UFLP) *is a special case of the universal FLP in which all facility cost functions are of the following form: for each $i \in \mathcal{F}$, $f_i(k) = 0$ if $k = 0$, and $f_i(k) = f_i$ if $k > 0$, where f_i is a constant which is called the facility cost of i.*

Definition 3 *The* metric soft-capacitated facility location problem (SCFLP) *is a special case of the universal FLP in which all facility cost functions are of the form $f_i(k) = f_i \lceil k/u_i \rceil$, where f_i and u_i are constants for every $i \in \mathcal{F}$, and u_i is called the capacity of facility i.*

The algorithms presented in this paper build upon an earlier approximation algorithm of Jain, Mahdian, and Saberi [13, 12], which is sketched below. We denote this algorithm by the JMS 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.

The analysis of the JMS algorithm has the feature that allows the approximation factor for the facility cost to be different from the approximation factor for the connection cost, and gives a way to compute the tradeoff between these two factors. The following definition captures this notion.

Definition 4 *An algorithm is called a (γ_f, γ_c)-approximation algorithm for the universal FLP, if for every instance $\mathcal{I}$ of the universal FLP, 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 $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.*

Recall the following theorem of Jain et al. [13, 12] on the approximation factor of the JMS algorithm.

Theorem A [13, 12]. *Let $\gamma_f \geq 1$ be fixed and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program which is referred to as the factor-revealing LP.*

$$\text{maximize} \quad \frac{\sum_{i=1}^{k} \alpha_i - \gamma_f f}{\sum_{i=1}^{k} d_i} \tag{LP1}$$

$$\begin{aligned}
\text{subject to} \quad & \forall\, 1 \leq i < k: \ \alpha_i \leq \alpha_{i+1} && (1)\\
& \forall\, 1 \leq j < i < k: \ r_{j,i} \geq r_{j,i+1} && (2)\\
& \forall\, 1 \leq j < i \leq k: \ \alpha_i \leq r_{j,i} + d_i + d_j && (3)\\
& \forall\, 1 \leq i \leq k: \ \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f && (4)\\
& \forall\, 1 \leq j \leq i \leq k: \ \alpha_j, d_j, f, r_{j,i} \geq 0 && (5)
\end{aligned}$$

Then the JMS algorithm is a (γ_f, γ_c)-approximation algorithm for UFLP. Furthermore, for $\gamma_f = 1$ we have $\gamma_c \leq 2$.

3 The uncapacitated facility location algorithm

3.1 The description of the algorithm

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. The technique of cost scaling has been previously used by Charikar and Guha [4] for the facility location problem in order to take advantage of the asymmetry between the performance of the algorithm with respect to the facility and connection costs. Here we give a different intuitive reason: Intuitively, the facilities that are opened by the JMS algorithm with the scaled-up facility costs are those that are very economical, because we weigh the facility cost more than the connection cost in the objective function. Therefore, we open these facilities in the first phase of the algorithm.

One important property of the JMS 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. [16, 12].

However, the facility costs have been scaled up in the first phase of our algorithm. Therefore, it is possible that the total cost (in terms of the original cost) can be reduced by opening an unopened facility that by reconnecting each city to its closest open facility. This motivates the second phase of our algorithm.

In the *second* phase of the algorithm, we decrease the scaling factor δ at rate 1, so at time t, the cost of facility i has reduced to $(\delta - t)f_i$. If at any point during this process, a facility could be opened without increasing the total cost (i.e., if 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. We stop when the scaling factor becomes 1. This is equivalent to a greedy procedure introduced by Guha and Khuller [8] and Charikar and Guha [4]. 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'_u, the ratio $(C - C'_u - f_u)/f_u$ is maximized. If this ratio is positive, then we open the facility u, and iterate; otherwise we stop. It is not hard to see that the second phase of our algorithm is equivalent to the Charikar-Guha-Khuller procedure: in the second phase of our algorithm, the first facility u that is opened corresponds to the minimum value of t, or the maximum value of $\delta - t$, for which we have $(\delta - t)f_u = C - C'_u$. In other words, our algorithm picks the facility u for which the value of $(C - C'_u)/f_u$ is maximized, and stops when this value becomes less than or equal to 1 for all u. This is the same as what the Charikar-Guha-Khuller procedure does. The original analysis of our algorithm in [18] was based on a lemma by Charikar and Guha [4]. Here we give an alternative analysis of our algorithm that only uses a single factor-revealing LP.

We denote our two-phase algorithm by algorithm A. In the remainder of this section, we analyze algorithm A, and prove that it always outputs a solution to the uncapacitated facility location problem of cost at most 1.52 times the optimum. The analysis is divided into three parts. First, in Section 3.2, we derive the factor-revealing linear program whose solution gives the approximation ratio of our algorithm. Next, in Section 3.3, we analyze this linear program, and compute its solution in terms of the approximation factors of the JMS algorithm. This gives the following result.

Theorem 1 *Let (γ_f, γ_c) be a pair obtained from the factor-revealing LP (LP1). Then for every $\delta \geq 1$, algorithm A is a $(\gamma_f + \ln(\delta) + \epsilon, 1 + \frac{\gamma_c - 1}{\delta})$-approximation algorithm for UFLP.*

Finally, we analyze the factor-revealing LP (LP1), and show that the JMS algorithm is a $(1.11, 1.78)$-approximation algorithm for UFLP. This, together with the above theorem for $\delta = 1.504$, implies that algorithm A is a 1.52-approximation algorithm for UFLP. We will show in Section 3.4 that this algorithm can be implemented in quasi-linear time, both for the distance oracle model and for the sparse graph model.

3.2 Deriving the factor-revealing LP

Recall that the JMS algorithm, in addition to finding a solution for the scaled instance, outputs the *share* of each city in the total cost of the solution. Let α_j denote the share of city j in the total cost. In other words,

α_j is the value of the variable B_j at the end of the JMS algorithm. Therefore the total cost of the solution is $\sum_j \alpha_j$. Consider an arbitrary collection $\mathcal{S}$ consisting of a single facility $f_{\mathcal{S}}$ and k cities. Let δf (f in the original instance) denote the opening cost of facility $f_{\mathcal{S}}$; α_j denote the share of city j in the total cost (where cities are ordered such that $\alpha_1 \leq \cdots \leq \alpha_k$); d_j denote the connection cost between city j and facility $f_{\mathcal{S}}$; and $r_{j,i}$ ($i > j$) denote the connection cost between city j and the facility that it is connected to at time α_i, right before city i gets connected for the first time (or if cities i and j get connected at the same time, define $r_{j,i} = \alpha_i = \alpha_j$). The main step in the analysis of the JMS algorithm is to prove that for any such collection $\mathcal{S}$, the δf, d_j, α_j, and $r_{j,i}$ values constitute a feasible solution to the program (LP1), where f is now replaced with δf since the facility costs have been scaled up by δ.

We implement and analyze the second phase as the following. Instead of decreasing the scaling factor continuously from δ to 1, we decrease it discretely in L steps where L is a constant. Let δ_i denote the value of the scaling factor in the i'th step. Therefore, $\delta = \delta_1 > \delta_2 > \ldots > \delta_L = 1$. We will fix the value of the δ_i's later. After decreasing the scaling factor from δ_{i-1} to δ_i, we consider facilities in an *arbitrary* order, and open those that can be opened without increasing the total cost. We denote this modified algorithm by A_L. Clearly, if L is sufficiently large (depending on the instance), the algorithm A_L computes the same solution as algorithm A.

In order to analyze the above algorithm, we need to add extra variables and inequalities to the inequalities in the factor-revealing program (LP1) given in Theorem A. Let $r_{j,k+i}$ denote the connection cost that city j in $\mathcal{S}$ pays after we change the scaling factor to δ_i and process all facilities as described above (Thus, $r_{j,k+1}$ is the connection cost of city j after the first phase). Therefore, by the description of the algorithm, we have

$$\forall\, 1 \leq i \leq L: \sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) \leq \delta_i f.$$

This is because if the above inequality is violated and if $f_{\mathcal{S}}$ is not open, we could open $f_{\mathcal{S}}$ and decrease the total cost. If $f_{\mathcal{S}}$ is open, then $r_{j,k+i} \leq d_j$ for all j and the inequality holds.

Now, we compute the share of the city j in the total cost of the solution that algorithm A_L finds. In the first phase of the algorithm, the share of city j in the total cost is α_j. Of this amount, $r_{j,k+1}$ is spent on the connection cost, and $\alpha_j - r_{j,k+1}$ is spent on the facility costs. However, since the facility costs are scaled up by a factor of δ in the first phase, therefore the share of city j in the *facility costs* in the original instance is equal to $(\alpha_j - r_{j,k+1})/\delta$. After we reduce the scaling factor from δ_i to δ_{i+1} ($i = 1, \ldots, L-1$), the connection cost of city j is reduced from $r_{j,k+i}$ to $r_{j,k+i+1}$. Therefore, in this step, the share of city j in the facility costs is $r_{j,k+i} - r_{j,k+i+1}$ with respect to the scaled instance, or $(r_{j,k+i} - r_{j,k+i+1})/\delta_{i+1}$ with respect to the original instance. Thus, at the end of the algorithm, the total share of city j in the facility costs is

$$\frac{\alpha_j - r_{j,k+1}}{\delta} + \sum_{i=1}^{L-1} \frac{r_{j,k+i} - r_{j,k+i+1}}{\delta_{i+1}}.$$

We also know that the final amount that city j pays for the connection cost is $r_{j,k+L}$. Therefore, the share of the city j in the total cost of the solution is:

$$\frac{\alpha_j - r_{j,k+1}}{\delta} + \sum_{i=1}^{L-1} \frac{r_{j,k+i} - r_{j,k+i+1}}{\delta_{i+1}} + r_{j,k+L+1} = \frac{\alpha_j}{\delta} + \sum_{i=1}^{L-1} \left(\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}\right) r_{j,k+i}. \tag{6}$$

This, together with a *dual fitting* argument similar to [12], implies the following.

Theorem 2 *Let (ξ_f, ξ_c) be such that $\xi_f \geq 1$ and ξ_c is an upper bound on the solution of the following maximization program for every k.*

$$\text{maximize} \quad \frac{\sum_{j=1}^{k}\left(\frac{\alpha_j}{\delta} + \sum_{i=1}^{L-1}\left(\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}\right) r_{j,k+i}\right) - \xi_f f}{\sum_{i=1}^{k} d_i} \quad \text{(LP2)}$$

$$\text{subject to} \quad \forall\, 1 \leq i < k: \ \alpha_i \leq \alpha_{i+1} \quad (7)$$

$$\forall\, 1 \leq j < i < k: \ r_{j,i} \geq r_{j,i+1} \quad (8)$$

$$\forall\, 1 \leq j < i \leq k: \ \alpha_i \leq r_{j,i} + d_i + d_j \quad (9)$$

$$\forall\, 1 \leq i \leq k: \ \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq \delta f \quad (10)$$

$$\forall\, 1 \leq i \leq L: \ \sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) \leq \delta_i f \quad (11)$$

$$\forall\, 1 \leq j \leq i \leq k: \ \alpha_j, d_j, f, r_{j,i} \geq 0 \quad (12)$$

Then, algorithm A_L is a (ξ_f, ξ_c)-approximation algorithm for UFLP.

3.3 Analyzing the factor-revealing LP

In the following theorem, we analyze the factor-revealing LP (LP2) and prove Theorem 1. In order to do this, we need to set the values of δ_i's. Here, for simplicity of computations, we set δ_i to $\delta^{\frac{L-i}{L-1}}$; however, it is easy to observe that any choice of δ_i's such that $\delta = \delta_1 > \delta_2 > \ldots > \delta_L = 1$ and the limit of $\max_i(\delta_i - \delta_{i+1})$ as L tends to infinity is zero, will also work.

Theorem 3 *Let (γ_f, γ_c) be a pair given by the maximization program (LP1) in Theorem A, and $\delta \geq 1$ be an arbitrary number. Then for every ϵ, if L is a sufficiently large constant, algorithm A_L is a $(\gamma_f + \ln(\delta) + \epsilon, 1 + \frac{\gamma_c - 1}{\delta})$-approximation algorithm for UFLP.*

Proof. Since the inequalities of the factor-revealing program (LP2) are a superset of the inequalities of the factor-revealing program (LP1), by Theorem A and the definition of (γ_f, γ_c), we have

$$\sum_{j=1}^{k} \alpha_j \leq \gamma_f \delta f + \gamma_c \sum_{j=1}^{k} d_j \quad (13)$$

By inequality (11), for every $i = 1, \ldots, L$, we have

$$\sum_{j=1}^{k} r_{j,k+i} \leq \sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) + \sum_{j=1}^{k} d_j \leq \delta_i f + \sum_{j=1}^{k} d_j. \quad (14)$$

Therefore,

$$\sum_{j=1}^{k}\left(\frac{\alpha_j}{\delta} + \sum_{i=1}^{L-1}\left(\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}\right) r_{j,k+i}\right)$$

$$
\begin{aligned}
&= \frac{1}{\delta}(\sum_{j=1}^{k}\alpha_j) + \sum_{i=1}^{L-1}\left((\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i})\sum_{j=1}^{k} r_{j,k+i}\right) \\
&\le \frac{1}{\delta}(\gamma_f \delta f + \gamma_c \sum_{j=1}^{k} d_j) + \sum_{i=1}^{L-1}\left((\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i})(\delta_i f + \sum_{j=1}^{k} d_j)\right) \\
&= \gamma_f f + \frac{\gamma_c}{\delta}\sum_{j=1}^{k} d_j + \sum_{i=1}^{L-1}(\frac{\delta_i}{\delta_{i+1}} - 1) f + (\frac{1}{\delta_L} - \frac{1}{\delta_1})\sum_{j=1}^{k} d_j \\
&= \left(\gamma_f + (L-1)(\delta^{1/(L-1)} - 1)\right) f + (\frac{\gamma_c}{\delta} + 1 - \frac{1}{\delta})\sum_{j=1}^{k} d_j.
\end{aligned}
$$

This, together with Theorem 2, shows that A_L is a $(\gamma_f + (L-1)(\delta^{1/(L-1)} - 1), 1 + \frac{\gamma_c - 1}{\delta})$-approximation algorithm for UFLP. The fact that the limit of $(L-1)(\delta^{1/(L-1)} - 1)$ as L tends to infinity is $\ln(\delta)$ completes the proof. ■

We observe that the proof of Theorem 3 goes through as long as the limit of $\sum_{i=1}^{L-1}(\frac{\delta_i}{\delta_{i+1}} - 1)$ as L tends to infinity is $\ln(\delta)$. This condition holds if we choose δ_i's such that $\delta = \delta_1 > \delta_2 > \ldots > \delta_L = 1$ and the limit of $\max_i(\delta_i - \delta_{i+1})$ as L tends to infinity is zero. It can be seen as follows. Let $x_i = \frac{\delta_i}{\delta_{i+1}} - 1 > 0$. Then, for $i = 1, 2, \cdots, L-1$,

$$x_i - o(x_i) \le \ln(\frac{\delta_i}{\delta_{i+1}}) \le x_i.$$

It follows that,

$$\sum_{i=1}^{L-1} x_i(1 - o(x_i)/x_i) \le \ln(\delta) \le \sum_{i=1}^{L-1} x_i.$$

Since $\lim_{L\to\infty}\frac{o(x_i)}{x_i} = \lim_{x_i\to 0}\frac{o(x_i)}{x_i} = 0$, we conclude that $\lim_{L\to\infty}\sum_{i=1}^{L-1} x_i = \ln(\delta)$.

Now we analyze the factor-revealing LP (LP1) and show that the JMS algorithm is a $(1.11, 1.78)$-approximation algorithm.

Lemma 4 *Let $\gamma_f = 1.11$. Then for every k, the solution of the factor-revealing LP (LP1) is at most 1.78.*

Proof. See Appendix. ■

Remark 1 *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 the above lemma for the value of γ_c is close to its actual value.*

3.4 Running time

The above analysis of the algorithm A, together with a recent result of Thorup [24], enables us to prove the following result.

Corollary 5 *For every $\epsilon > 0$, there is a quasi-linear time $(1.52 + \epsilon)$-approximation algorithm for UFLP, both in the distance oracle model and in the sparse graph model.*

Proof Sketch. We use the algorithm A_L for a large constant L. Thorup [24] shows that for every $\epsilon > 0$, the JMS algorithm can be implemented in quasi-linear time (in both the distance oracle and the sparse graph models) with an approximation factor of $1.61 + \epsilon$. It is straightforward to see that his argument actually implies the stronger conclusion that the quasi-linear algorithm is a $(\gamma_f + \epsilon, \gamma_c + \epsilon)$-approximation, where (γ_f, γ_c) are given by Theorem A. This shows that the first phase of algorithm A_L can be implemented in quasi-linear time. The second phase consists of a constant number of rounds. Therefore, we only need to show that each of these rounds can be implemented in quasi-linear time. This is easy to see in the distance oracle model. In the sparse graph model, we can use the exact same argument as the one used by Thorup in the proof of Lemma 5.1 of [24]. ∎

4 The linear-cost facility location problem

The *linear-cost facility location problem* is a special case of the universal FLP in which the facility costs are of the form

$$f_i(k) = \begin{cases} 0 & k = 0 \\ a_i k + b_i & k > 0 \end{cases}$$

where a_i and b_i are nonnegative values for each $i \in \mathcal{F}$. a_i and b_i are called the marginal (a.k.a. incremental) and setup cost of facility i, respectively.

We denote an instance of the linear-cost FLP with marginal costs (a_i), setup costs (b_i), and connection costs (c_{ij}) by $LFLP(a, b, c)$. Clearly, the regular UFLP is a special case of the linear-cost FLP with $a_i = 0$, i.e., $LFLP(0, b, c)$. Furthermore, it is straightforward to see that $LFLP(a, b, c)$ is equivalent to an instance of the regular UFLP in which the marginal costs are added to the connection costs. More precisely, let $\bar{c}_{ij} = c_{ij} + a_i$ for $i \in \mathcal{F}$ and $j \in \mathcal{C}$, and consider an instance of UFLP with facility costs (b_i) and connection costs $(\bar{c}_{ij})$. We denote this instance by $UFLP(b, c + a)$. It is easy to see that $LFLP(a, b, c)$ is equivalent to $UFLP(b, c + a)$. Thus, the linear-cost FLP can be solved using any algorithm for UFLP, and the overall approximation ratio will be the same. However, for applications in the next section, we need bifactor approximation factors of the algorithm (as defined in Definition 4).

It is not necessarily true that applying a (γ_f, γ_c)-approximation algorithm for UFLP on the instance $UFLP(b, a+c)$ will give a (γ_f, γ_c)-approximate solution for $LFLP(a, b, c)$. However, we will show that the JMS algorithm has this property. The following lemma generalizes Theorem A for the linear-cost FLP.

Lemma 6 *Let (γ_f, γ_c) be a pair obtained from the factor-revealing LP in Theorem A. Then applying the JMS algorithm on the instance $UFLP(b, a+c)$ will give a (γ_f, γ_c)-approximate solution for $LFLP(a, b, c)$.*

Proof. Let SOL be an arbitrary solution for $LFLP(a, b, c)$, which can also be viewed as a solution for $UFLP(b, \bar{c})$ for $\bar{c} = c + a$. Consider a facility f that is open in SOL, and the set of clients connected to it in SOL. Let k denote the number of these clients, $f(k) = ak + b$ (for $k > 0$) be the facility cost function of f, and $\bar{d}_j$ denote the connection cost between client j and the facility f in the instance $UFLP(b, a + c)$. Therefore, $d_j = \bar{d}_j - a$ is the corresponding connection cost in the original instance $LFLP(a, b, c)$. Recall the definition of α_j and $r_{j,i}$ in the factor-revealing LP of Theorem A. By inequality (3) we also know that $\alpha_i \le r_{j,i} + \bar{d}_j + \bar{d}_i$. We strengthen this inequality as follows.

Claim 7 $\alpha_i \le r_{j,i} + d_j + d_i$

Proof. It is true if $\alpha_i = \alpha_j$ since it happens only if $r_{j,i} = \alpha_j$. Otherwise, consider clients i and $j(< i)$ at time $t = \alpha_i - \epsilon$. Let s be the facility j is assigned to at time t. By triangle inequality, we have

$$\bar{c}_{si} = c_{si} + a_s \le c_{sj} + d_i + d_j + a_s = \bar{c}_{sj} + d_i + d_j \le r_{j,i} + d_i + d_j.$$

On the other hand $\alpha_i \le \bar{c}_{si}$ since otherwise i could have connected to facility s at a time earlier than t. ■

Also, by inequality (4) we know that

$$\sum_{j=1}^{i-1} \max(r_{j,i} - \bar{d}_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - \bar{d}_j, 0) \le b.$$

Notice that $\max(a - x, 0) \ge \max(a, 0) - x$ if $x \ge 0$. Therefore, we have

$$\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le b + ka. \tag{15}$$

Claim 7 and Inequality (15) show that the values α_j, $r_{j,i}$, d_j, a, and b constitute a feasible solution of the following optimization program.

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

However, it is clear that the above optimization program and the factor-revealing LP in Theorem A are equivalent. This completes the proof of this lemma. ■

The above lemma and Theorem A give us the following corollary, which will be used in the next section.

Corollary 8 *There is a* $(1, 2)$*-approximation algorithm for the linear-cost facility location problem.*

It is worth mentioning that algorithm A can also be generalized for the linear-cost FLP. The only trick is to scale up both a and b in the first phase by a factor of δ, and scale them both down in the second phase. The rest of the proof is almost the same as the proof of Lemma 6.

5 The soft-capacitated facility location problem

In this section we will show how the soft-capacitated facility location problem can be reduced to the linear-cost FLP. In Section 5.1 we define the concept of reduction between facility location problems. We will use this concept in Sections 5.2 and 5.3 to obtain approximation algorithms for SCFLP and a generalization of SCFLP and the concave-cost FLP.

5.1 Reduction between facility location problems

Definition. A reduction from a facility location problem $\mathcal{A}$ to another facility location problem $\mathcal{B}$ is a polynomial-time procedure R that maps every instance $\mathcal{I}$ of $\mathcal{A}$ to an instance $R(\mathcal{I})$ of $\mathcal{B}$. This procedure is called a (σ_f, σ_c)-reduction if the following conditions hold.

1. For any instance $\mathcal{I}$ of $\mathcal{A}$ and any feasible solution for $\mathcal{I}$ with facility cost $F^*_{\mathcal{A}}$ and connection cost $C^*_{\mathcal{A}}$, there is a corresponding solution for the instance $R(\mathcal{I})$ with facility cost $F^*_{\mathcal{B}} \leq \sigma_f F^*_{\mathcal{A}}$ and connection cost $C^*_{\mathcal{B}} \leq \sigma_c C^*_{\mathcal{A}}$.
2. For any feasible solution for the instance $R(\mathcal{I})$, there is a corresponding feasible solution for $\mathcal{I}$ whose total cost is at most as much as the total cost of the original solution for $R(\mathcal{I})$. In other words, cost of the instance $R(\mathcal{I})$ is an over-estimate of cost of the instance $\mathcal{I}$.

Theorem 9 *If there is a (σ_f, σ_c)-reduction from a facility location problem $\mathcal{A}$ to another facility location problem $\mathcal{B}$, and a (γ_f, γ_c)-approximation algorithm for $\mathcal{B}$, then there is a $(\gamma_f\sigma_f, \gamma_c\sigma_c)$-approximation algorithm for $\mathcal{A}$.*

Proof. On an instance $\mathcal{I}$ of the problem $\mathcal{A}$, we compute $R(\mathcal{I})$, run the (γ_f, γ_c)-approximation algorithm for $\mathcal{B}$ on $R(\mathcal{I})$, and output the corresponding solution for $\mathcal{I}$. In order to see why this is a $(\gamma_f\sigma_f, \gamma_c\sigma_c)$-approximation algorithm for $\mathcal{A}$, let SOL denote an arbitrary solution for $\mathcal{I}$, ALG denote the solution that the above algorithm finds, and $F^*_{\mathcal{P}}$ and $C^*_{\mathcal{P}}$ ($F^{ALG}_{\mathcal{P}}$ and $C^{ALG}_{\mathcal{P}}$, respectively) denote the facility and connection costs of SOL (ALG, respectively) when viewed as a solution for the problem $\mathcal{P}$ ($\mathcal{P} = \mathcal{A}, \mathcal{B}$). By the definition of (σ_f, σ_c)-reductions and (γ_f, γ_c)-approximation algorithms we have

$$F^{ALG}_{\mathcal{A}} + C^{ALG}_{\mathcal{A}} \leq F^{ALG}_{\mathcal{B}} + C^{ALG}_{\mathcal{B}} \leq \gamma_f F^*_{\mathcal{B}} + \gamma_c C^*_{\mathcal{B}} \leq \gamma_f\sigma_f F^*_{\mathcal{A}} + \gamma_c\sigma_c C^*_{\mathcal{A}},$$

which completes the proof of the lemma. ■

We will see examples of reductions in the rest of this paper.

5.2 The soft-capacitated facility location problem

In this subsection, we give a 2-approximation algorithm for the soft-capacitated FLP by reducing it to the linear-cost FLP.

Theorem 10 *There is a 2-approximation algorithm for the soft-capacitated facility location problem.*

Proof. We use the following reduction: Construct an instance of the linear-cost FLP, where we have the same sets of facilities and clients. The connection costs remain the same. However, the facility cost of the ith facility is $(1+\frac{k-1}{u_i})f_i$ if $k \geq 1$ and 0 if $k=0$. Note that, for every $k \geq 1$, $\lceil \frac{k}{u_i} \rceil \leq 1+\frac{k-1}{u_i} \leq 2 \cdot \lceil \frac{k}{u_i} \rceil$. Therefore, it is easy to see that this reduction is a $(2,1)$-reduction. By Lemma 8, there is a $(1,2)$-approximation algorithm for the linear-cost FLP, which together with Theorem 9 completes the proof. ■

Furthermore, we now illustrate that the following natural linear programming formulation of SCFLP has an integrality gap of 2. This means that we cannot obtain a better approximation ratio using this LP relaxation as the lower bound.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}x_{ij} \\
\text{subject to} \quad & \forall\, i\in\mathcal{F}, j\in\mathcal{C}:\; x_{ij}\leq y_i \\
& \forall\, i\in\mathcal{F}:\; \sum_{j\in\mathcal{C}} x_{ij} \leq u_i y_i \\
& \forall\, j\in\mathcal{C}:\; \sum_{i\in\mathcal{F}} x_{ij} = 1 \\
& \forall\, i\in\mathcal{F}, j\in\mathcal{C}:\; x_{ij}\in\{0,1\} && (16)\\
& \forall\, i\in\mathcal{F}:\; y_i \text{ is a nonnegative integer} && (17)
\end{aligned}$$

In a natural linear program relaxation, we replace the constraints (16) and (17) by $x_{ij} \geq 0$ and $y_i \geq 0$. Here we see that even if we only relax constraint (17), the integrality gap is 2. Consider an instance of SCFLP that consists of only one potential facility i, and $k \geq 2$ clients. Assume that the capacity of facility i is $k-1$, the facility cost is 1, and all connection costs are 0. It is clear that the optimal integral solution has cost 2. However, after relaxing constraint (17), the optimal fractional solution has cost $1+\frac{1}{k-1}$. Therefore, the integrality gap between the integer program and its relaxation is $\frac{2(k-1)}{k}$ which tends to 2 as k tends to infinity.

5.3 The concave soft-capacitated facility location problem

In this subsection, we consider a common generalization of the soft-capacitated facility location problem and the concave-cost facility location problem. This problem, which we refer to as the *concave soft-capacitated FLP*, is the same as the soft-capacitated FLP except that if $r \geq 0$ copies of facility i are open, then the facility cost is $g_i(r)a_i$ where $g_i(r)$ is a given concave increasing function of r. In other words, the concave soft-capacitated FLP is a special case of the universal FLP in which the facility cost functions are of the form $f_i(x) = a_i g_i(\lceil x/u_i \rceil)$ for constants a_i, u_i and a concave increasing function g_i. It is also a special case of the so-called stair-case cost facility location problem [11]. On the other hand, it is a common generalization of the soft-capacitated FLP (when $g_i(r) = r$) and the concave-cost FLP (when $u_i = 1$ for all i). The concave-cost FLP is a special case of the universal FLP in which facility cost functions are required to be concave and increasing (See [9]). The main result of this subsection is the following.

Theorem 11 *The concave soft-capacitated FLP is* $(\max_{i\in\mathcal{F}} \frac{g_i(2)}{g_i(1)}, 1)$*-reducible to the linear-cost FLP.*

The above theorem is established by the following lemmas which show the reductions between the concave soft-capacitated FLP, the concave-cost FLP and the linear-cost FLP. Notice that $\max_{i\in\mathcal{F}} \frac{g_i(2)}{g_i(1)} \leq 2$.

Lemma 12 *The concave soft-capacitated FLP is* $(\max_{i\in\mathcal{F}} \frac{g_i(2)}{g_i(1)}, 1)$ *reducible to the concave-cost FLP.*

Proof. Given an instance $\mathcal{I}$ of the concave soft-capacitated FLP, where the facility cost function of the facility i is $f_i(k) = g_i(\lceil k/u_i \rceil)a_i$, we construct an instance $R(\mathcal{I})$ of the concave-cost FLP as follows: We have the same sets of facilities and clients and the same connection costs as in $\mathcal{I}$. The facility cost function of the ith facility is given by

$$f'_i(k) = \begin{cases} \Big(g_i(r) + (g_i(r+1) - g_i(r))(\frac{k-1}{u_i} - r + 1)\Big) a_i & \text{if } k > 0,\ r := \lceil k/u_i \rceil \\ 0 & \text{if } k = 0. \end{cases}$$

Concavity of g_i implies that the above function is also concave, and therefore $R(\mathcal{I})$ is an instance of concave-cost FLP. Also, it is easy to see from the above definition that

$$g_i(\lceil k/u_i \rceil)a_i \leq f'_i(k) \leq g_i(\lceil k/u_i \rceil + 1)a_i.$$

By the concavity of the function g_i, we have $\frac{g_i(r+1)}{g_i(r)} \leq \frac{g_i(2)}{g_i(1)}$ for every $r \geq 1$. Therefore, for every facility i and number k,

$$f_i(k) \leq f'_i(k) \leq \frac{g_i(2)}{g_i(1)} f_i(k).$$

This completes the proof of the lemma. ■

Now, we will show a simple $(1,1)$-reduction from the concave-cost FLP to the linear-cost FLP. This, together with the above lemma, reduces the concave soft-capacitated facility location problem to the linear-cost FLP.

Lemma 13 *There is a* $(1,1)$*-reduction from the concave-cost FLP to the linear-cost FLP.*

Proof. Given an instance $\mathcal{I}$ of concave-cost FLP, we construct an instance $R(\mathcal{I})$ of linear-cost FLP as follows: Corresponding to each facility i in $\mathcal{I}$ with facility cost function $f_i(k)$, we put n copies of this facility in $R(\mathcal{I})$ (where n is the number of clients), and let the facility cost function of the l'th copy be

$$f_i^{(l)}(k) = \begin{cases} f_i(l) + (f_i(l) - f_i(l-1))(k-l) & \text{if } k > 0 \\ 0 & \text{if } k = 0. \end{cases}$$

In other words, the facility cost function is the line that passes through the points $(l-1, f(l-1))$ and $(l, f(l))$. The set of clients, and the connection costs between clients and facilities are unchanged. We prove that this reduction is a $(1,1)$-reduction.

For any feasible solution SOL for $\mathcal{I}$, we can construct a feasible solution SOL' for $R(\mathcal{I})$ as follows: If a facility i is open and k clients are connected to it in SOL, we open the k'th copy of the corresponding facility in $R(\mathcal{I})$, and connect the clients to it. Since $f_i(k) = f_i^{(k)}(k)$, the facility and connection costs of SOL' is the same as those of SOL.

Conversely, consider an arbitrary feasible solution SOL for $R(\mathcal{I})$. We construct a solution SOL' for $\mathcal{I}$ as follows. For any facility i, if at least one of the copies of i is open in SOL, we open i and connect all clients that were served by a copy of i in SOL to it. We show that this does not increase the total cost of the solution: Assume the l_1'th, l_2'th, ..., and l_s'th copies of i were open in SOL, serving $k_1, k_2, \ldots$, and k_s clients, respectively. By concavity of f_i, and the fact that $f_i^{(l)}(k) \geq f_i^{(k)}(k) = f_i(k)$ for every l, we have

$$f_i(k_1 + \cdots + k_s) \leq f_i(k_1) + \cdots + f_i(k_s) \leq f_i^{(l_1)}(k_1) + \cdots + f_i^{(l_s)}(k_s).$$

This shows that the facility cost of SOL' is at most the facility cost of SOL. ■

6 Conclusion

We have obtained the best approximation ratios for two well-studied facility location problems, 1.52 for UFLP and 2 for SCFLP, respectively. The approximation ratio for UFLP almost matches the lower bound of 1.463, and the approximation ratio for SCFLP achieves the integrality gap of the standard LP relaxation of the problem. An interesting open question in this area is to close the gap between 1.52 and 1.463 for UFLP.

Although the performance guarantee of our algorithm for UFLP is very close to the lower bound of 1.463, it would be nice to show that the bound of 1.52 is actually tight. In [12], it was shown that a solution to the factor-revealing LP for the JMS algorithm provides a tight bound on the performance guarantee of the JMS algorithm. It is reasonable to expect that a solution to (LP2) may also be used to construct a tight example for our 1.52-approximation algorithm. However, we were unsuccessful in constructing such an example.

Our results (Theorem 1 and Lemma 4) for UFLP and/or the idea of bifactor reduction have been used to get the currently best known approximations ratios for several multi-level facility location problems [1, 26]. Since UFLP is the most basic facility location problem, we expect to see more applications of our results.

Acknowledgments. We would like to thank Asaf Levin for pointing out that our analysis of the 2-approximation algorithm for the soft-capacitated facility location problem is tight. We also like to mention that an idea to derive better approximation factors for UFLP using the $(1, 2)$ bifactor guarantee was independently proposed earlier by Kamal Jain in a private communication to the first author, and by the last two authors. We thank the anonymous referees for their helpful suggestions that significantly improved the exposition of our paper.

References

[1] A. Ageev, Y. Ye, and J. Zhang. Improved combinatorial approximation algorithms for the k-level facility location problem. *SIAM J Discrete Math.*, 18(1):207–217, 2004.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal of Computing*, 33(3):544–562, 2004.

[3] P. Bauer and R. Enders. A capacitated facility location problem with integer decision variables. In *International Symposium on Mathematical Programming (ISMP)*, 1997.

[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] 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.

[6] F.A. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2003.

[7] 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.

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

[9] M. Hajiaghayi, M. Mahdian, and V.S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1):42–47, August 2003.

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

[11] K. Holmberg. Solving the staircase cost facility location problem with decomposition and piecewise linearization. *European Journal of Operational Research*, 74:41–61, 1994.

[12] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. Approximation algorithms for facility location via dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6):795–824, November 2003.

[13] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the 34st Annual ACM Symposium on Theory of Computing*, pages 731–740, 2002.

[14] 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.

[15] M.R. Korupolu, C.G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000.

[16] 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.

[17] M. Mahdian and M. Pál. Universal facility location. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA)*, 2003.

[18] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Proceedings of 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2002)*, volume 2462 of *Lecture Notes in Computer Science*, pages 229–242, 2002.

[19] M. Mahdian, Y. Ye, and J. Zhang. A 2-approximation algorithm for the soft-capacitated facility location problem. In *Proceedings of 6th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2003)*, volume 2764 of *Lecture Notes in Computer Science*, pages 129–140, 2003.

[20] C.S. Revell and G. Laporte. The plant location problem: New models and research prospects. *Operations Research*, 44:864–874, 1996.

[21] 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.

[22] 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.

[23] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization*, pages 240–257, 2002.

[24] M. Thorup. Quick and good facility location. In *Proceedings of the 14th ACM-SIAM symposium on Discrete Algorithms*, 2003.

[25] M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.

[26] J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, 2004.

A Proof of Lemma 4

Proof. By doubling a feasible solution of the factor-revealing program (LP1) (as in the proof of Lemma 12 in [13]) it is easy to show that for every k, $z_k \leq z_{2k}$. Therefore, without loss of generality, we can assume that k is sufficiently large.

Consider a feasible solution of the factor-revealing LP. Let $x_{j,i} := \max(r_{j,i} - d_j, 0)$. Inequality (4) of the factor-revealing LP implies that for every $i \leq i'$,

$$(i' - i + 1)\alpha_i - f \leq \sum_{j=i}^{i'} d_j - \sum_{j=1}^{i-1} x_{j,i}. \tag{18}$$

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

$$(l_i - i + 1)\alpha_i - f \leq \sum_{j=i}^{l_i} d_j - \sum_{j=1}^{i-1} x_{j,i}. \tag{19}$$

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.

$$\theta_i = \begin{cases} \frac{1}{l_i - i + 1} & \text{if } i \leq p_3 k \\ \frac{1}{(1-p_3)k} & \text{if } p_3 k < i \leq (1 - p_3)k \\ \frac{p_4 k}{(k-i)(k-i+1)} & \text{if } (1 - p_3)k < i \leq p_2 k \\ 0 & \text{if } i > p_2 k \end{cases} \tag{20}$$

By multiplying both sides of inequality (19) 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.

$$\sum_{i=1}^{p_1 k} \alpha_i - (\sum_{i=1}^{p_1 k} \theta_i) f \le \sum_{i=1}^{p_1 k} \sum_{j=i}^{p_2 k} \frac{d_j}{p_2 k - i + 1} - \sum_{i=1}^{p_1 k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{p_2 k - i + 1} \tag{21}$$

$$\sum_{i=p_1 k+1}^{p_3 k} \alpha_i - (\sum_{i=p_1 k+1}^{p_3 k} \theta_i) f \le \sum_{i=p_1 k+1}^{p_3 k} \sum_{j=i}^{k} \frac{d_j}{k - i + 1} - \sum_{i=p_1 k+1}^{p_3 k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{k - i + 1} \tag{22}$$

$$\sum_{i=p_3 k+1}^{(1-p_3)k} \frac{k-i+1}{(1-p_3)k} \alpha_i - (\sum_{i=p_3 k+1}^{(1-p_3)k} \theta_i) f \le \sum_{i=p_3 k+1}^{(1-p_3)k} \sum_{j=i}^{k} \frac{d_j}{(1-p_3)k} - \sum_{i=p_3 k+1}^{(1-p_3)k} \sum_{j=1}^{i-1} \frac{\max(r_{j,i} - d_j, 0)}{(1-p_3)k} \tag{23}$$

$$\begin{aligned} \sum_{i=(1-p_3)k+1}^{p_2 k} \frac{p_4 k}{k-i} \alpha_i - (\sum_{i=(1-p_3)k+1}^{p_2 k} \theta_i) f \le & \sum_{i=(1-p_3)k+1}^{p_2 k} \sum_{j=i}^{k} \frac{p_4 k d_j}{(k-i)(k-i+1)} \\ & - \sum_{i=(1-p_3)k+1}^{p_2 k} \sum_{j=1}^{i-1} \frac{p_4 k \max(r_{j,i} - d_j, 0)}{(k-i)(k-i+1)} \end{aligned} \tag{24}$$

We define $s_i := \max_{l \ge i} (\alpha_l - d_l)$. Using this definition and inequalities (2) and (3) of the factor-revealing LP (LP1) we obtain

$$\forall i: \; r_{j,i} \ge s_i - d_j \Longrightarrow \forall i: \; \max(r_{j,i} - d_j, 0) \ge \max(s_i - 2d_j, 0) \tag{25}$$

$$\forall i: \; \alpha_i \le s_i + d_i \tag{26}$$

$$s_1 \ge s_2 \ge \ldots \ge s_k \; (\ge 0) \tag{27}$$

We assume $s_k \ge 0$ here because that, if on contrary $\alpha_k < d_k$, we can always set α_k equal to d_k without violating any constraint in the factor-revealing LP (LP1) and increase z_k.

Inequality (26) and $p_4 \le 1 - p_2$ imply

$$\begin{aligned} & \sum_{i=p_3 k+1}^{(1-p_3)k} \left(1 - \frac{k-i+1}{(1-p_3)k}\right) \alpha_i + \sum_{i=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-i}\right) \alpha_i + \sum_{i=p_2 k+1}^{k} \alpha_i \\ \le & \sum_{i=p_3 k+1}^{(1-p_3)k} \frac{i - p_3 k - 1}{(1-p_3)k} (s_i + d_i) + \sum_{i=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-i}\right) (s_i + d_i) + \sum_{i=p_2 k+1}^{k} (s_i + d_i) \end{aligned} \tag{28}$$

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

$$\begin{aligned} \zeta &= \sum_{i=1}^{p_1 k} \frac{1}{p_2 k - i + 1} + \sum_{i=p_1 k+1}^{p_3 k} \frac{1}{k-i+1} + \sum_{i=p_3 k+1}^{(1-p_3)k} \frac{1}{(1-p_3)k} + \sum_{i=(1-p_3)k+1}^{p_2 k} \left(\frac{p_4 k}{k-i} - \frac{p_4 k}{k-i+1}\right) \\ &= \ln\left(\frac{p_2}{p_2 - p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} + o(1). \end{aligned} \tag{29}$$

By adding the inequalities (21), (22), (23), (24), (28) and using (25), (27), 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_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{s_i - 2d_j}{2(p_2 k - i + 1)} \\
& + \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{s_i - 2d_j}{k - i + 1} \\
& + \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{s_i - 2d_j}{(1-p_3)k} \\
& + \sum_{i=(1-p_3)k+1}^{p_2 k} \sum_{j=i}^{k} \frac{p_4 k d_j}{(k-i)(k-i+1)} - \sum_{i=(1-p_3)k+1}^{p_2 k} \sum_{j=1}^{i-1} \frac{p_4 k \max(s_{p_2 k+1} - 2d_j, 0)}{(k-i)(k-i+1)} \\
& + \sum_{i=p_3 k+1}^{(1-p_3)k} \frac{i - p_3 k - 1}{(1-p_3)k}(s_i + d_i) + \sum_{i=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-i}\right)(s_i + d_i) + \sum_{i=p_2 k+1}^{k} (s_{p_2 k+1} + d_i) \\
= \; & \sum_{j=1}^{p_2 k} \sum_{i=1}^{\min(j, p_1 k)} \frac{d_j}{p_2 k - i + 1} - \sum_{i=1}^{p_1 k} \frac{i-1}{2(p_2 k - i + 1)} s_i + \sum_{j=1}^{p_1 k} \sum_{i=j+1}^{p_1 k} \frac{d_j}{p_2 k - i + 1} \\
& + \sum_{j=p_1 k+1}^{k} \sum_{i=p_1 k+1}^{\min(j, p_3 k)} \frac{d_j}{k - i + 1} - \sum_{i=p_1 k+1}^{p_3 k} \frac{i-1}{k-i+1} s_i + \sum_{j=1}^{p_3 k} \sum_{i=\max(j, p_1 k)+1}^{p_3 k} \frac{2d_j}{k - i + 1} \\
& + \sum_{j=p_3 k+1}^{k} \sum_{i=p_3 k+1}^{\min(j,(1-p_3)k)} \frac{d_j}{(1-p_3)k} - \sum_{i=p_3 k+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_3 k)+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_2 k)} \left(\frac{1}{k-i} - \frac{1}{k-i+1}\right) p_4 k d_j \\
& - \sum_{j=1}^{p_2 k} \sum_{i=\max(j,(1-p_3)k)+1}^{p_2 k} p_4 k \left(\frac{1}{k-i} - \frac{1}{k-i+1}\right) \max(s_{p_2 k+1} - 2d_j, 0) \\
& + \sum_{i=p_3 k+1}^{(1-p_3)k} \frac{i - p_3 k - 1}{(1-p_3)k}(s_i + d_i) + \sum_{i=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-i}\right)(s_i + d_i) + \sum_{i=p_2 k+1}^{k} d_i \\
& + (1-p_2) k s_{p_2 k+1}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \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} \\
\leq \quad & \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\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\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
\end{aligned}
$$

$$-\sum_{j=1}^{p_3k}\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\max(s_{p_2k+1}-2d_j,0)-\sum_{j=p_3k+1}^{(1-p_3)k}\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)(s_{p_2k+1}-2d_j)$$
$$-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{p_3k}{(1-p_3)k}s_j$$
$$+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)s_j+(1-p_2)ks_{p_2k+1} \tag{30}$$

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

$$\begin{aligned}
&\sum_{i=1}^{k}\alpha_i-\zeta f\\
\leq\;&\sum_{j=1}^{k}\lambda_jd_j-\sum_{j=1}^{p_1k}\frac{j-1}{2(p_2k-j+1)}s_j-\sum_{j=p_1k+1}^{p_3k}\frac{j-1}{k-j+1}s_j-\sum_{j=p_3k+1}^{(1-p_3)k}\frac{p_3k}{(1-p_3)k}s_j\\
&+\sum_{j=(1-p_3)k+1}^{p_2k}\left(1-\frac{p_4k}{k-j}\right)s_j+\left(1-p_2-(1-2p_3)\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\right)ks_{p_2k+1}\\
&-\left(\frac{p_4}{1-p_2}-\frac{p_4}{p_3}\right)\sum_{j=1}^{p_3k}\max(s_{p_2k+1}-2d_j,0),
\end{aligned} \tag{31}$$

where

$$\lambda_j:=\begin{cases}
\ln(\frac{p_2}{p_2-p_1})+2\ln(\frac{1-p_1}{1-p_3})+\frac{2(1-2p_3)}{1-p_3}+o(1) & \text{if } 1\leq j\leq p_1k\\
\ln(\frac{p_2}{p_2-p_1})+\ln(\frac{1-p_1}{1-p_3})+\frac{2(1-2p_3)}{1-p_3}+\mathrm{H}_{k-j}-\mathrm{H}_{(1-p_3)k}+o(1) & \text{if } p_1k<j\leq p_3k\\
\ln(\frac{p_2}{p_2-p_1})+\ln(\frac{1-p_1}{1-p_3})+\frac{2(1-2p_3)}{1-p_3}+\frac{2p_4}{1-p_2}-\frac{2p_4}{p_3}+o(1) & \text{if } p_3k<j\leq(1-p_3)k\\
\ln(\frac{p_2}{p_2-p_1})+\ln(\frac{1-p_1}{1-p_3})+\frac{1-2p_3}{1-p_3}+1-\frac{p_4}{p_3}+o(1) & \text{if } (1-p_3)k<j\leq p_2k\\
\ln(\frac{1-p_1}{1-p_3})+\frac{1-2p_3}{1-p_3}+1+\frac{p_4}{1-p_2}-\frac{p_4}{p_3}+o(1) & \text{if } p_2k<j\leq k.
\end{cases}$$

For every $j\leq p_3k$, we have

$$\lambda_{(1-p_3)k}-\lambda_j\leq\frac{2p_4}{1-p_2}-\frac{2p_4}{p_3}\Rightarrow\delta_j:=(\lambda_{(1-p_3)k}-\lambda_j)\Big/\left(\frac{2p_4}{1-p_2}-\frac{2p_4}{p_3}\right)\leq 1. \tag{32}$$

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

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

then for every $j \leq p_3 k$, $\lambda_j \leq \lambda_{(1-p_3)k}$ and therefore $\delta_j \geq 0$. Then, since $0 \leq \delta_j \leq 1$, we can replace $\max(s_{p_2k+1} - 2d_j, 0)$ by $\delta_j(s_{p_2k+1} - 2d_j)$ in (31). This gives us

$$
\begin{aligned}
& \sum_{i=1}^{k} \alpha_i - \zeta f \\
\leq \;& \sum_{j=1}^{k} \lambda_j d_j - \sum_{j=1}^{p_1 k} \frac{j-1}{2(p_2 k - j + 1)} s_j - \sum_{j=p_1 k+1}^{p_3 k} \frac{j-1}{k-j+1} s_j - \sum_{j=p_3 k+1}^{(1-p_3)k} \frac{p_3 k}{(1-p_3)k} s_j \\
& + \sum_{j=(1-p_3)k+1}^{p_2 k} \left(1 - \frac{p_4 k}{k-j}\right) s_j + \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k s_{p_2 k+1} \\
& - \frac{1}{2} \sum_{j=1}^{p_3 k} (\lambda_{(1-p_3)k} - \lambda_j)(s_{p_2k+1} - 2d_j) \qquad (34)
\end{aligned}
$$

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

$$
\sum_{i=1}^{k} \alpha_i - \zeta f \leq \lambda_{(1-p_3)k} \sum_{j=1}^{(1-p_3)k} d_j + \sum_{j=(1-p_3)k+1}^{k} \lambda_j d_j + \sum_{j=1}^{p_2 k+1} \mu_j s_j, \qquad (35)
$$

where

$$
\mu_j = \begin{cases} -\dfrac{j-1}{2(p_2 k - j + 1)} & \text{if } 1 \leq j \leq p_1 k \\ -\dfrac{j-1}{k-j+1} & \text{if } p_1 k < j \leq p_3 k \\ -\dfrac{p_3}{1-p_3} & \text{if } p_3 k < j \leq (1-p_3)k \\ 1 - \dfrac{p_4 k}{k-j} & \text{if } (1-p_3)k < j \leq p_2 k \end{cases} \qquad (36)
$$

and

$$
\begin{aligned}
& \mu_{p_2 k+1} \\
= \;& \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k - \frac{1}{2}\lambda_{(1-p_3)k} p_3 k + \frac{1}{2}\sum_{j=1}^{p_3 k} \lambda_j \\
= \;& \left(1 - p_2 - (1-2p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right)\right) k - \frac{1}{2}\lambda_{(1-p_3)k} p_3 k \\
& + \frac{p_1 k}{2}\left(\ln(\frac{p_2}{p_2 - p_1}) + 2\ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right) \\
& + \frac{(p_3 - p_1)k}{2}\left(\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + o(1)\right) + \frac{1}{2}\sum_{j=p_1 k+1}^{p_3 k} \sum_{i=(1-p_3)k+1}^{k-j} \frac{1}{i} \\
= \;& \left(\ln(\frac{1-p_1}{1-p_3}) + 2 - 2p_2 - p_3 + p_1 - 2(1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right) + o(1)\right)\frac{k}{2} \qquad (37)
\end{aligned}
$$

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.,

$$\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{2(1-2p_3)}{1-p_3} + \frac{2p_4}{1-p_2} - \frac{2p_4}{p_3} < \gamma \tag{38}$$

$$\ln(\frac{p_2}{p_2-p_1}) + \ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 - \frac{p_4}{p_3} < \gamma \tag{39}$$

and

$$\ln(\frac{1-p_1}{1-p_3}) + \frac{1-2p_3}{1-p_3} + 1 + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < \gamma. \tag{40}$$

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

$$\mu_1 + \mu_2 + \cdots + \mu_i \leq 0, \tag{41}$$

then by inequality (27), 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

$$\ln\left(\frac{p_2}{p_2-p_1}\right) + \ln\left(\frac{1-p_1}{1-p_3}\right) + \frac{1-2p_3}{1-p_3} + \frac{p_4}{1-p_2} - \frac{p_4}{p_3} < 1.11, \tag{42}$$

then inequality (35) can be written as

$$\sum_{i=1}^{k} \alpha_i - 1.11f \leq \gamma \sum_{j=1}^{k} d_j, \tag{43}$$

which shows that the solution of the maximization program (LP1) is at most γ. From (36), it is clear that $\mu_j \leq 0$ for every $j \leq (1-p_3)k$ and $\mu_j \geq 0$ for every $(1-p_3)k \leq j \leq p_2 k$. Therefore, it is enough to check inequality (41) for $i = p_2 k$ and $i = p_2 k + 1$. We have

$$\begin{aligned}
\sum_{j=1}^{p_2k} \mu_j &= -\sum_{j=1}^{p_1k} \frac{p_2k - p_2k + j - 1}{2(p_2k - j + 1)} - \sum_{j=p_1k+1}^{p_3k} \frac{k - k + j - 1}{k - j + 1} - \frac{p_3(1-2p_3)k}{1-p_3} \\
&\quad + (p_2 - 1 + p_3)k - \sum_{j=(1-p_3)k+1}^{p_2k} \frac{p_4 k}{k-j} \\
&= -\frac{p_2k}{2}(\mathrm{H}_{p_2k} - \mathrm{H}_{(p_2-p_1)k}) + \frac{p_1k}{2} - k(\mathrm{H}_{(1-p_1)k} - \mathrm{H}_{(1-p_3)k}) + (p_3 - p_1)k \\
&\quad - \frac{p_3(1-2p_3)k}{1-p_3} + (p_2 - 1 + p_3)k - p_4k(\mathrm{H}_{p_3k} - \mathrm{H}_{(1-p_2)k}) \\
&= \Bigg(-\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2-p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} \\
&\quad - p_4 \ln(\frac{p_3}{1-p_2}) + o(1)\Bigg) k
\end{aligned} \tag{44}$$

Therefore, inequality (41) is equivalent to the following two inequalities.

$$-\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2 - p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4 \ln(\frac{p_3}{1-p_2}) < 0 \tag{45}$$

$$\begin{aligned} -\frac{p_1}{2} + p_2 + 2p_3 - 1 - \frac{p_2}{2}\ln(\frac{p_2}{p_2 - p_1}) - \ln(\frac{1-p_1}{1-p_3}) - \frac{p_3(1-2p_3)}{1-p_3} - p_4 \ln(\frac{p_3}{1-p_2}) \\ +\frac{1}{2}\ln(\frac{1-p_1}{1-p_3}) + 1 - p_2 - \frac{p_3}{2} + \frac{p_1}{2} - (1-p_3)\left(\frac{p_4}{1-p_2} - \frac{p_4}{p_3}\right) < 0 \end{aligned} \tag{46}$$

Now, it is enough to observe that if we let $p_1 = 0.225, p_2 = 0.791, p_3 = 0.30499$, $p_4 = 0.06984$, and $\gamma = 1.7764$, then $p_1 < p_3 < 1 - p_3 < p_2$ and $p_4 < 1 - p_2$ as specified earlier, and inequalities (33), (38), (39), (40), (42), (45), and (46) are all satisfied. Therefore, the solution of the optimization program (LP1) is at most $1.7764 < 1.78$. ■