Math. Program., Ser. A 108, 159–176 (2006)

Digital Object Identifier (DOI) 10.1007/s10107-006-0704-x

Jiawei Zhang

Approximating the two-level facility location problem via a quasi-greedy approach*

Received: October 3, 2003 / Accepted: January 13, 2006
Published online: 6 March 2006 – © Springer-Verlag 2006

Abstract. We propose a *quasi-greedy* algorithm for approximating the classical uncapacitated 2-level facility location problem (2-LFLP). Our algorithm, unlike the standard greedy algorithm, selects a sub-optimal candidate at each step. It also relates the minimization 2-LFLP problem, in an interesting way, to the maximization version of the single level facility location problem. Another feature of our algorithm is that it combines the technique of randomized rounding with that of dual fitting.

This new approach enables us to approximate the metric 2-LFLP in polynomial time with a ratio of 1.77, a significant improvement on the previously known approximation ratios. Moreover, our approach results in a local improvement procedure for the 2-LFLP, which is useful in improving the approximation guarantees for several other multi-level facility location problems. An additional result of our approach is an $O(\ln(n))$-approximation algorithm for the non-metric 2-LFLP, where n is the number of clients. This is the first non-trivial approximation for a non-metric multi-level facility location problem.

Key words. Two-level facility location – Approximation algorithm – Linear programming relaxation – Quasi-greedy approach

1. Introduction

1.1. Problem statement

In the single level uncapacitated facility location problem, we are given a set of clients and a set of facilities. We want to open a subset of the facilities such that all the clients are served by the open facilities and the total cost of opening facilities and serving clients is minimized. In the k-level uncapacitated facility location problem (k-LFLP), the demands must be routed among facilities in a hierarchical order, i.e., from the highest level (the factories) down to the lowest (the retailers), before reaching the clients. The k-LFLP arises naturally in designing logistic systems.

The k-LFLP can be formulated formally as follows. We are given a set of clients $\mathcal{D}$ and k level sets of facilities $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_k$. Denote $\mathcal{P} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_k$ and $\mathcal{F} = \cup_{t=1}^{k} \mathcal{F}_t$. Each client $j \in D$ must be served by an open path $p = (i_1, i_2, \cdots, i_k) \in \mathcal{P}$ of k facilities with exactly one facility from each of the k levels, where a path p is open if and only if every facility on the path is open. There is a facility cost f_{i_t} for opening facility $i_t \in \mathcal{F}_t$ $(1 \le t \le k)$. Furthermore, If client $j \in \mathcal{D}$ is served

J. Zhang: IOMS-Operations Management, Stern School of Business, New York University. 44 W. 4th Street, Suite 8-66, New York, NY 10012-1126, USA. e-mail: jzhang@stern.nyu.edu.

* An extended abstract of this paper appeared in the *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), January 2004.*

by an open path $p = (i_1, i_2, \cdots, i_k) \in \mathcal{P}$ a connection cost c_{jp} is incurred where $c_{jp} = c_{ji_1} + \sum_{t=2}^{k} c_{i_{t-1}i_t}$ and c_{ji} is the connection cost between j and i for $j, i \in \mathcal{D} \cup \mathcal{F}$. Here, we wish to open a subset of facilities such that each client is assigned to an open path and the total cost is minimized, i.e., to choose $\emptyset \neq S_t \subset \mathcal{F}_t$, $t = 1, 2, \cdots, k$, such that

$$\sum_{j \in D} \min_{p \in S_1 \times S_2 \times \cdots \times S_k} c_{jp} + \sum_{t=1}^{k} \sum_{i_t \in S_t} f_{i_t}$$

is minimized. We also assume that the connection costs are nonnegative, symmetric, and satisfy the triangle inequality, i.e., for each $i, j, l \in \mathcal{D} \cup \mathcal{F}$, $c_{ij} \geq 0$, $c_{ij} = c_{ji}$ and $c_{ij} \leq c_{il} + c_{lj}$.

In this paper we are concerned with the 2-LFLP, the most studied special case of k-LFLP in the literature of Operations Research for $k \geq 2$ [1, 8, 20, 21, 24, 33–35, 38]. The study on the 2-LFLP is motivated by the fact that in many applications, especially in supply chains, there are hierarchical two-level structures. The problem also has applications in telecommunications and computer network designs [8]. On the other hand, although it is the simplest model among all the k-LFLP for $k \geq 2$, the 2-LFLP has some fundamental structural differences from the 1-LFLP. For example, the 2-LFLP does not possess the so-called supermodularity, a well-known property for the 1-LFLP [25]. This property is often helpful in designing branch-and-cut algorithms and in analyzing some approximation algorithms. Thus, the 2-LFLP needs new techniques.

1.2. Previous results

The 2-LFLP generalizes the popular single level uncapacitated facility location problem (1-LFLP), which is already NP-hard [15]. Since the work of Shmoys, Tardos and Aardal [37], designing approximation algorithms for the 1-LFLP and its related problems has received considerable attention during the past few years [36]. We call an algorithm of a minimization problem a $\rho(\geq 1)$-approximation algorithm if for any instance of the problem the algorithm runs in polynomial time and outputs a solution that has a cost at most ρ times the minimal cost, where ρ is called the performance guarantee or the approximation ratio of the algorithm. Guha and Khuller [22] proved that the existence of a polynomial time 1.463-approximation algorithm for the 1-LFLP would imply that $P = NP$. And the best currently known approximation ratio for the 1-LFLP is 1.52 due to Mahdian, Ye and Zhang [30]. Therefore, the approximability of the 1-LFLP is well understood.

However, the 2-LFLP remains intriguing. One can easily see that the lower bound 1.463 of the 1-LFLP also applies to the 2-LFLP, and no better lower bound is known. In [37], the algorithm for the 1-LFLP has been extended to the 2-LFLP with an approximation ratio 3.16. Later on, Aardal, Chudak and Shmoys [2] showed that the k-LFLP can be approximated in polynomial time by a factor of 3 for any $k \geq 2$ using a linear programming relaxation. However, their algorithm does not possess a better performance guarantee for $k = 2$, neither do a series of recently proposed faster combinatorial algorithms due to Meyerson, Munagala, and Plotkin [32], Guha, Meyerson, and Munagala

[27], Bumb and Kern [7], and Ageev [3]. In fact, the algorithms of [2, 7, 3] will produce solutions whose open paths are disjoint, and Edwards [17] showed that such algorithms can not have worst case ratios that are better than 3 even for $k = 2$. Very recently, Ageev, Ye, and Zhang [5] proposed two different combinatorial algorithms and showed that 'the better of the two' has a performance guarantee 2.43, although each of them has a performance guarantee at least 3.

1.3. Our results and techniques

The main result of this paper is that the 2-LFLP can be approximated in polynomial time by a factor of 1.77, a significant improvement over the previous performance guarantees, since no one can do better than 1.463 unless $P = NP$. The improved ratio is achieved by using what we call *a quasi-greedy approach*. Our algorithm is analyzed by using the technique of a factor-revealing LP developed by Jain, Mahdian, and Saberi [28]. One advantage of our algorithm is that it can be easily generalized to solve the so-called two level concentrator location problem (see [10] and the references therein) with the same ratio 1.77. The quasi-greedy approach also results in a local improvement procedure for the 2-LFLP, which does not improve the ratio 1.77 for the 2-LFLP but is useful in improving the performance guarantees for other multi-level facility location problems. In particular, we show that 3- and the 4-LFLP can be approximated by factors of 2.51 and 2.81, respectively, obtaining the current best approximation ratios for these two problems. We also obtain an improved 3-approximation algorithm for the 2-LFLP with soft capacities. An additional result of our approach is an $O(\ln(|\mathcal{D}|))$-approximation algorithm for the non-metric 2-LFLP where the connection cost may not satisfy the triangle inequality. This is the first non-trivial approximation algorithm for a non-metric multi-level facility location problem. And its approximation ratio matches the best known approximation ratio for the non-metric 1-LFLP due to Hochbaum [23]. Furthermore, the approximation ratio is the best possible up to a constant, unless P=NP [19]. We remark that the set covering formulation for the two-level facility location problem does not yield a direct logarithmic bound, since determining the cheapest set per newly covered client is a non-trivial problem.

Greedy algorithms have been successful in tackling the 1-LFLP [28, 30]. In a standard greedy approach, at each step, one computes a greedy function value for each element of a candidate set and chooses the optimal candidate based on these values. When applied to the 1-LFLP, the candidate set is the set of (unopen) facilities, and the greedy function is the ratio of the cost incurred to the number of new clients served, which can be computed easily.

However, the issue is more complicated when we apply the greedy approach to the 2-LFLP. In particular, it is difficult to define a candidate set. Some straightforward choices do not work such as $\mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{F}_1 \times \mathcal{F}_2$. One sophisticated definition works where a candidate is defined as many facilities with exactly one in $\mathcal{F}_2$ and the rest in $\mathcal{F}_1$. Unfortunately, the problem is that there are exponentially many candidates, and we can not choose the best candidate by comparing their greedy function values with each other in polynomial time.

Thus, we make a simple but important observation for the 2-LFLP: choosing the best candidate among the exponentially many candidates according to an 'easy' greedy function is equivalent to choosing the best candidate among polynomially many candidates according to an appropriately defined ‘hard’ greedy function. In the latter, given a candidate, it is NP-hard to compute the exact value for the ‘hard’ greedy function. But the good news is that we may compute the greedy function value approximately in polynomial time. Therefore, we could choose the ‘best’ candidate according to the approximated greedy function value. We call this approach *quasi-greedy* since it may not choose the ‘greediest’ candidate.

It turns out that computing the ‘hard’ greedy function value for solving the 2-LFLP is equivalent to the so-called maximization version of the 1-LFLP (Max-1-LFLP in short). Recall that in the minimization 1-LFLP, assigning a client to an open facility will incur a connection cost. In the Max-1-LFLP, a revenue will be generated by assigning a client to an open facility, and the objective is to maximize the net profit (the total revenue minus the total facility cost). The Max-1-LFLP and the minimization 1-LFLP are equivalent from the perspective of optimization, but not from that of approximation. Approximation algorithms for the Max-1-LFLP have a longer history than those for the 1-LFLP[14, 4]. However, the results of [14, 4] do not help in establishing our result for the 2-LFLP. We shall give a new approximation algorithm for the Max-1-LFLP such that the approximation result can be used in proving our bound for the 2-LFLP. To the best of our knowledge, this is the first time that approximations for the maximization and the minimization versions of a facility location problem have been related.

We remark that the quasi-greedy approach has been used in designing approximation algorithms in other settings; see, for example, Chekuri et al. [13] and Charikar et al. [11]. However, our approach is new in that it reduces the size of the set of candidates from exponentially many to polynomially many in such a way that the greedy function for the latter can be well approximated.

Our algorithm and analysis combine the technique of randomized rounding with that of dual fitting. Both techniques have been used for solving various facility location problems, but never combined before. In our algorithm, each greedy step is solved by randomly rounding the solution of its linear programming relaxation. Dual fitting (and the factor-revealing LP) is used for proving the overall performance guarantee.

1.4. Organization of the paper

The rest of the paper is organized as follows. In Section 2, we present a linear programming based approximation algorithm for the Max-1-LFLP. The quasi-greedy algorithm for the 2-LFLP and its analysis is given in Section 3. In Section 4, a local improvement procedure based on the quasi-greedy approach for the 2-LFLP is proposed. In section 5, we present improved approximation algorithms for the non-metric 2-LFLP, the two level concentrator location problem, and other multi-level facility location problems. Final remarks are given in Section 6.

2. An algorithm for the Max-1-LFLP

In this section, we consider the maximization version of the (single level) facility location problem (Max-1-LFLP). In the Max-1-LFLP, we are given the set of client $\mathcal{D}$, the set of facilities $\mathcal{F}$. The facility cost for opening facility i is f_i and the revenue generated by assigning client j to facility i is $d_{ij} \geq 0$. Here d_{ij} might not satisfy the triangle inequality. The objective is to open a subset of the facilities of $\mathcal{F}$ and then assign each of the clients in $\mathcal{D}$ to an open facility such that the net profit is maximized. The Max-1-LFLP can be formulated as the following integer program.

$$\begin{aligned} \text{Max} \quad & \sum_{i \in \mathcal{F}, j \in \mathcal{D}} d_{ij} x_{ij} - \sum_{i \in \mathcal{F}} f_i y_i \\ \text{s.t.} \quad & \sum_{i \in \mathcal{F}} x_{ij} \leq 1 \quad \text{for all} \quad j \in \mathcal{D} \\ & x_{ij} \leq y_i \quad \text{for all} \quad i \in \mathcal{F}, j \in \mathcal{D} \\ & x_{ij}, y_i \in \{0, 1\} \quad \text{for all} \quad i \in \mathcal{F}, j \in \mathcal{D} \end{aligned} \tag{1}$$

where $y_i = 1$ if we decide to open facility i, otherwise $y_i = 0$; $x_{ij} = 1$ if client j is assigned to facility i, otherwise $x_{ij} = 0$.

Consider any feasible solution, whose total profit is assumed to be $C^* - F^*$, where C^* and F^* correspond to the total revenue and the total facility cost, respectively. Our goal is to find an algorithm that produces a solution with profit $C - F \geq \left(1 - \frac{1}{e}\right) C^* - F^*$. The algorithm is based on the linear programming (LP) relaxation of this problem. However, the actual LP we will solve is different from the exact LP relaxation of (1).

Algorithm MAX

Step 1. Solve the following LP and obtain an optimal solution (x, y):

$$\begin{aligned} \text{Max} \quad & \left(1 - \frac{1}{e}\right) \cdot \sum_{i \in \mathcal{F}, j \in \mathcal{D}} d_{ij} x_{ij} - \sum_{i \in \mathcal{F}} f_i y_i \\ \text{s.t.} \quad & \sum_{i \in \mathcal{F}} x_{ij} \leq 1 \quad \text{for all} \quad j \in \mathcal{D} \\ & x_{ij} \leq y_i \quad \text{for all} \quad i \in \mathcal{F}, j \in \mathcal{D} \\ & 0 \leq x_{ij}, y_i \leq 1 \quad \text{for all} \quad i \in \mathcal{F}, j \in \mathcal{D} \end{aligned} \tag{2}$$

Step 2. For each $i \in \mathcal{F}$, open facility i independently with probability y_i, and assign each client $j \in \mathcal{D}$ to an open facility with maximum revenue. Let the resulting solution be $(\hat{x}, \hat{y})$ and the corresponding facility cost and revenue be F and C respectively.

We are ready to present the main result of this section.

Theorem 1. *There exists an algorithm that finds a solution with profit $C - F$ such that*

$$C - F \geq \left(1 - \frac{1}{e}\right) C^* - F^*.$$

Proof. By the definition of $(\hat{x}, \hat{y})$, $\hat{y}_i$ is 1 with probability y_i, and 0 with probability $1 - y_i$. Then $\mathrm{E}\left[\hat{y}_i\right] = y_i$ and thus the expected facility cost is

$$\mathrm{E}[F] = \mathrm{E}\left[\sum_{i \in \mathcal{F}} f_i \hat{y}_i\right] = \sum_{i \in \mathcal{F}} f_i \mathrm{E}\left[\hat{y}_i\right] = \sum_{i \in \mathcal{F}} f_i y_i.$$

It is left to analyze the quantity

$$\mathrm{E}[C] = \mathrm{E}\left[\sum_{i \in \mathcal{F}, j \in \mathcal{D}} d_{ij} \hat{x}_{ij}\right].$$

Consider any $j \in \mathcal{D}$, and assume that the facilities are indexed such that $d_{1j} \leq d_{2j} \leq \cdots \leq d_{|\mathcal{F}|j}$. For convenience, we define $d_{0j} = 0$. Now, for $i = 1, 2, \cdots, |\mathcal{F}|$, let X_i be an indicator variable that is 1 if some facility from the set $i, i+1, \cdots, |\mathcal{F}|$ is open and is 0 otherwise. Therefore, the revenue generated by client j can be expressed as

$$\sum_{i \in \mathcal{F}} d_{ij} \hat{x}_{ij} = \sum_{i=1}^{|\mathcal{F}|} X_i \left(d_{ij} - d_{(i-1)j}\right).$$

Consequently,

$$\mathrm{E}\left[\sum_{i \in \mathcal{F}} d_{ij} \hat{x}_{ij}\right] = \sum_{i=1}^{|\mathcal{F}|} \mathrm{E}[X_i] \left(d_{ij} - d_{(i-1)j}\right). \tag{3}$$

Now we observe that, by using the inequality $1 - x \leq e^{-x}$,

$$\Pr[X_i = 0] = \prod_{k=i}^{|\mathcal{F}|} (1 - y_k) \leq \prod_{k=i}^{|\mathcal{F}|} \exp(-y_k) \leq \exp\left(-\sum_{k=i}^{|\mathcal{F}|} y_k\right).$$

Moreover, since $x_{kj} \leq y_k$ for $k = 1, 2, \cdots, |\mathcal{F}|$, we have

$$\exp\left(-\sum_{k=i}^{|\mathcal{F}|} y_k\right) \leq \exp\left(-\sum_{k=i}^{|\mathcal{F}|} x_{kj}\right).$$

It follows that

$$\mathrm{E}[X_i] = \Pr[X_i = 1] \geq 1 - \exp\left(-\sum_{k=i}^{|\mathcal{F}|} x_{kj}\right) \geq \left(1 - \frac{1}{e}\right) \sum_{k=i}^{|\mathcal{F}|} x_{kj}$$

where we use the fact that $1 - e^{-x} \geq \left(1 - e^{-1}\right) x$ holds for $0 \leq x \leq 1$. Then we get,

$$\begin{aligned}\sum_{i=1}^{|\mathcal{F}|} \mathrm{E}[X_i] \left(d_{ij} - d_{(i-1)j}\right) &\geq \sum_{i=1}^{|\mathcal{F}|} \left(1 - \frac{1}{e}\right) \sum_{k=i}^{|\mathcal{F}|} x_{kj} \left(d_{ij} - d_{(i-1)j}\right) \\ &= \left(1 - \frac{1}{e}\right) \sum_{i=1}^{|\mathcal{F}|} d_{ij} \left(\sum_{k=i}^{|\mathcal{F}|} x_{kj} - \sum_{k=i+1}^{|\mathcal{F}|} x_{kj}\right) \\ &= \left(1 - \frac{1}{e}\right) \sum_{i=1}^{|\mathcal{F}|} d_{ij} x_{ij},\end{aligned}$$

which, together with (3), implies that

$$\mathrm{E}\left[\sum_{i \in \mathcal{F}} d_{ij} \hat{x}_{ij}\right] \geq \left(1 - \frac{1}{e}\right) \sum_{i=1}^{|\mathcal{F}|} d_{ij} x_{ij}.$$

Therefore,

$$\mathrm{E}\left[\sum_{i \in \mathcal{F}, j \in \mathcal{D}} d_{ij} \hat{x}_{ij}\right] - \mathrm{E}\left[\sum_{i \in \mathcal{F}} f_i \hat{y}_i\right] \geq \left(1 - \frac{1}{e}\right) \sum_{i \in \mathcal{F}, j \in \mathcal{D}} d_{ij} x_{ij} - \sum_{i \in \mathcal{F}} f_i y_i.$$

Since (x, y) is the optimal solution for the linear program (2), the right hand side of the above inequality is bounded below by $\left(1 - \frac{1}{e}\right) C^* - F^*$. (Recall that in the objective function of (2), the d_{ij}'s are scaled by a factor of (1-1/e).) Therefore,

$$\mathrm{E}[C - F] \geq \left(1 - \frac{1}{e}\right) C^* - F^*.$$

By the standard technique using conditional expectation (see [18]), we can derandomize the algorithm such that

$$C - F \geq \left(1 - \frac{1}{e}\right) C^* - F^*.$$

□

3. The quasi-greedy algorithm for the 2-LFLP

For our purpose, it would be useful to have the following definition.

Definition 1. An algorithm is called a (R_f, R_c)-approximation algorithm for the k-LFLP, if for every instance $\mathcal{I}$ of the k-LFLP, 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 $R_f F^{SOL} + R_c C^{SOL}$.

We cite a lemma from Mahdian et al. [30].

Lemma 1. *[30] For* $(a,b) = (1.104, 1.78)$ *or* $(a,b) = (1.118, 1.77)$*,* $\sum_{i=1}^{n} \alpha_i \le a \cdot \sum_{i=1}^{n} m_i + b \cdot f$*, if the following system of inequalities holds*

$$\begin{aligned} &\forall\, 1 \le j < n: \ \alpha_j \le \alpha_{j+1} \\ &\forall\, 1 \le l < j < n: \ r_{l,j} \ge r_{l,j+1} \\ &\forall\, 1 \le l < j \le n: \ \alpha_j \le r_{l,j} + m_j + m_l \\ &\forall\, 1 \le j \le n: \ \sum_{l=1}^{j-1} \max(r_{l,j} - m_l, 0) + \sum_{l=j}^{n} \max(\alpha_j - m_l, 0) \le f \\ &\forall\, 1 \le l \le j \le n: \ \alpha_j, m_j, f, r_{l,j} \ge 0. \end{aligned} \tag{4}$$

Furthermore, the 1*-LFLP can be approximated by a factor of* $(a + ln(\delta), 1 + (b-1)/\delta)$ *for any* $\delta \ge 1$.

Below are the details of our quasi-greedy algorithm for the 2-LFLP. The algorithm is presented in a way similar to that of Jain et al. [28].

Algorithm QG

1. We introduce a notion of time. The algorithm starts at time 0. At this time, all clients are unconnected and all facilities are unopen. Let U be the set of unconnected clients. Thus at this time $U = \mathcal{D}$. And $\alpha_j = 0$ for each $j \in \mathcal{D}$.

At each moment, every client j will have some money B_j available to offer to each unopen facility in $\mathcal{F}_2$, where $B_j = \alpha_j$ if j is unconnected, and $B_j = c_{jp}$ if j is currently connected to an open path p. The amount of offers received by a facility $i \in \mathcal{F}_2$ is computed as follows. Consider any $i \in \mathcal{F}_2$. For each $j \in \mathcal{D}$ and $k \in \mathcal{F}_1$, define $d_{kj} = \max\{B_j - c_{jki}, 0\}$ where $c_{jki} = c_{jk} + c_{ki}$, and $\hat{f}_k = 0$ if $k \in \mathcal{F}_1$ is already open; $\hat{f}_k = f_k$ otherwise. Then we obtain an instance of the Max-1-LFLP. We solve this instance by Algorithm MAX and obtain a feasible solution. This profit (which could be negative) is the amount of offers received by the facility i. Note that each client j can make an offer to a facility $i \in \mathcal{F}_2$ through exactly one facility $\sigma_i(j) \in \mathcal{F}_1$ where $\sigma_i(j)$ is the facility to which j is assigned by Algorithm MAX. The contribution made by client j to facility i is equal to $d_{\sigma_i(j)j}$.

2. While $U \neq \emptyset$, increase the time and simultaneously increase α_j at the same rate for each $j \in U$, until one of the following events occurs:
   (1). For an unopen facility $i \in \mathcal{F}_2$, the total amount of offers that it has received from the clients is equal to f_i. In this case, we open facility i. And for each client $j \in \mathcal{D}$ who has made non-zero contribution to i, we open facility $\sigma_i(j) \in \mathcal{F}_1$ and assign j to the path $(\sigma_i(j), i)$. Furthermore, if $j \in U$, then remove j from U (and stop increasing α_j).
   (2). For a client $j \in U$ and open facilities $i \in \mathcal{F}_2$ and $k \in \mathcal{F}_1$, we have that $\alpha_j = c_{jki}$; then assign j to the path (k, i) and remove j from U (and stop increasing α_j).

Remark 1. For each client $j \in \mathcal{D}$, the value α_j will increase until j gets connected to an open path, and α_j will not change after that. At each moment, the value B_j is used to denote the amount of money available to client j to offer. Before j is connected,

$B_j = \alpha_j$. As soon as j is connected to a path, say p, for the first time, α_j will be fixed, and B_j will be set to c_{jp} (which could be strictly less than α_j). In general, after j is connected, B_j is the connection cost paid by client j. For example, client j may get connected later to another open path p' with $c_{jp'} \leq c_{jp}$ to save the connection cost. Then as long as j is connected to p', $B_j = c_{jp'}$. Therefore, the value B_j will not stop increasing until j gets connected to an open path (it may start decreasing from then on).

Remark 2. In order to implement Algorithm QG in polynomial time, we notice that the total number of possible events is bounded by $|\mathcal{D}| + |\mathcal{F}_2|$. At any time, we need to find the minimum value of how much the α_j's should increase such that the next event will occur. This can be done in polynomial time (but not strongly polynomial time) by performing a bisection search. Another way to implement the algorithm is that we discretize the time and only consider the values of α_j's that are powers of $(1+\epsilon)$, i.e., $\{0, 1, (1+\epsilon), (1+\epsilon)^2, (1+\epsilon)^3, \cdots, \}$ for any given constant $\epsilon > 0$. Therefore, the algorithm can be implemented in polynomial time for any given constant $\epsilon > 0$. Our analysis below is based on this implementation of the algorithm.

Now, we are ready to analyze the algorithm. First, we have

Lemma 2. *The total cost of the solution produced by Algorithm QG is* $\sum_{j\in\mathcal{D}} \alpha_j$.

We can assume that the open paths of an optimal solution of the 2-LFLP form a forest [17]. In order to analyze the performance guarantee, we consider any tree of the forest. The root of this tree must be a facility $i \in \mathcal{F}_2$. And the leaves of the tree are a subset of the facilities, say $S_i \subset \mathcal{F}_1$. We also consider the set of clients, denoted by D_i, who are assigned to the tree rooted at i in the optimal solution. Therefore, the total cost (of the optimal solution) associated with this tree is

$$f_i + \sum_{k\in S_i} f_k + \sum_{j\in D_i} \min_{k\in S_i} c_{jki}.$$

If we could prove that, for each $i \in \mathcal{F}_2$,

$$\sum_{j\in D_i} \alpha_j \leq R_f \cdot \left(f_i + \sum_{k\in S_i} f_k \right) + R_c \cdot \sum_{j\in D_i} \min_{k\in S_i} c_{jki},$$

then Algorithm QG must be a (R_f, R_c)-approximation algorithm for the 2-LFLP. Thus, in the rest of this section, we consider a particular i, and the associated S_i and D_i. We assume that $|D_i| = n$ and let $m_j = \min_{k\in S_i} c_{jki}$. Furthermore, without losing generality, we assume that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$.

For each $j : 1 \leq j \leq n$, consider the situation of the algorithm at time $t = \alpha_j$. For each $l \leq j-1$, if l is connected to a path p before time t (i.e., l was connected to a path at time $\alpha_j/(1+\epsilon)$), then let $r_{l,j} = c_{lp}$; otherwise, let $r_{l,j} = \alpha_l$. In the latter case, $\alpha_l = \alpha_j$.

Lemma 3. *Let* $f = \frac{e}{e-1}\left(f_i + \sum_{k \in S_i} f_k\right)$*; then the system of inequalities (5) holds:*

$$\begin{aligned}
&\forall\, 1 \le j < n : \ \alpha_j \le \alpha_{j+1};\\
&\forall\, 1 \le l < j < n : \ r_{l,j} \ge r_{l,j+1};\\
&\forall\, 1 \le l < j \le n : \ \alpha_j/(1+\epsilon) \le r_{l,j} + m_j + m_l;\\
&\forall\, 1 \le j \le n : \ \sum_{l=1}^{j-1} \max\left(r_{l,j}/(1+\epsilon) - m_l, 0\right)\\
&\qquad\qquad + \sum_{l=j}^{n} \max\left(\alpha_j/(1+\epsilon) - m_l, 0\right) \le f;\\
&\forall\, 1 \le l \le j \le n : \ \alpha_j, m_j, f, r_{l,j} \ge 0.
\end{aligned} \tag{5}$$

Proof. First of all, the inequality $\alpha_j \le \alpha_{j+1}$ holds by assumption. And for each $l : 1 \le l < j < n$, we have $r_{l,j} \ge r_{l,j+1}$ since once a client is connected to a path, it will never be connected to a path with a higher connection cost.

Consider clients $j > l$ at time $t = \alpha_j$. If l is unconnected at time $t/(1+\epsilon)$, then by definition, $r_{l,j} = \alpha_l$ and it must be the case $\alpha_j = \alpha_l$. Therefore, $\alpha_j \le r_{l,j} + m_j + m_l$. Now we assume that l is connected to a path p at time $t/(1+\epsilon)$. It follows that $r_{l,j} = c_{lp}$ and p must be open at time $t/(1+\epsilon)$. Thus $t/(1+\epsilon) \le c_{jp}$, otherwise j could be connected to p before time t. However, by triangle inequality, $c_{jp} \le c_{lp} + m_j + m_l$. Thus, we also have $\alpha_j/(1+\epsilon) = t/(1+\epsilon) \le r_{l,j} + m_j + m_l$.

It is left to prove that for all j,

$$\sum_{l=1}^{j-1} \max\left(r_{l,j}/(1+\epsilon) - m_l, 0\right) + \sum_{l=j}^{n} \max\left(\alpha_j/(1+\epsilon) - m_l, 0\right) \le \frac{e}{e-1}\left(f_i + \sum_{k \in S_i} f_k\right).$$

At time $t = \alpha_j/(1+\epsilon)$, Algorithm QG will construct an instance of the Max-1-LFLP, i.e., for each client $l \in \mathcal{D}$ and facility $k \in \mathcal{F}_1$, define $d_{kl} = \max\{B_l - c_{lki}, 0\}$. Note that $B_l \ge r_{l,j}/(1+\epsilon)$ for $l < j$ and $B_l = t$ for $l \ge j$. Then we consider a feasible solution of the Max-1-LFLP. We open facilities in S_i with opening cost at most $\sum_{k \in S_i} f_k$. For each $l \in D_i$, we assign l to the facility in S_i, to which l is connected in the optimal solution of the 2-LFLP. For other clients not in D_i, we assign them to any open facility. The total profit of this solution is at least

$$\sum_{l \in \mathcal{D}_i} \max\{B_l - m_l, 0\} - \sum_{k \in S_i} f_k.$$

By Theorem 1, Algorithm QG can find a solution for the Max-1-LFLP with total profit at least

$$\left(1 - \frac{1}{e}\right) \sum_{l \in \mathcal{D}_i} \max\{B_l - m_l, 0\} - \sum_{k \in S_i} f_k.$$

However, at time $t = \alpha_j/(1+\epsilon)$, the total amount of offers received by i from the clients is at least this quantity that can not be greater than f_i. Therefore,

$$\left(1 - \frac{1}{e}\right) \sum_{l \in \mathcal{D}_i} \max\{B_l - m_l, 0\} - \sum_{k \in S_i} f_k \le f_i.$$

Rearranging the terms in the above inequality, we get the desired conclusion. □

Theorem 2. *The* 2*-LFLP can be approximated by a factor of* $1.77(1+\epsilon)^2$ *in polynomial time for any given constant* $\epsilon > 0$.

Proof. By Lemma 1 and Lemma 3, by comparing systems (4) and (5), and by using the pair $(a, b) = (1.118, 1.77)$, we get

$$\sum_{j \in D_i} \frac{\alpha_j}{(1+\epsilon)^2} \le 1.118 \cdot \frac{e}{e-1} \left(f_i + \sum_{k \in S_i} f_k \right) + 1.77 \cdot \sum_{j \in D_i} m_i,$$

which, together with Lemma 2, implies that Algorithm QG is a $1.77(1+\epsilon)^2$-approximation algorithm for the 2-LFLP since $1.118 \cdot \frac{e}{e-1} \le 1.77$. □

4. A local improvement procedure for the 2-LFLP

The main result of this section is that given a (R_f, R_c)-approximation algorithm for the 2-LFLP, we can find an approximation algorithm with performance guarantee $\left(R_f + \frac{e}{e-1}\ln(\delta), 1 + \frac{R_c - 1}{\delta}\right)$ for any $\delta \ge 1$ in polynomial time. We will see its application in the analysis of algorithms for the 3-LFLP and the 4-LFLP, among others. A similar result has been proved for UFLP by Guha and Khuller [22] and we have seen many applications of it [12, 22, 30, 31, 5].

The key here is again the quasi-greedy approach. In order to prove the above result, we design a local improvement procedure for the 2-LFLP. Roughly speaking, once we have a feasible solution for the 2-LFLP, we may add some facilities to the current solution such that the total cost is reduced.

The local improvement procedure proceeds as follows, which is similar to Algorithm QG. We are given a solution for the 2-LFLP. Assume the connection cost of each client $j \in \mathcal{D}$ is o_j. We want to add some facilities (open one facility in $\mathcal{F}_2$ and simultaneously many facilities in $\mathcal{F}_1$) to the current solution. At each step, we consider an unopen facility $i \in \mathcal{F}_2$. We construct an instance of the Max-1-LFLP: we are given the set of clients $\mathcal{D}$, the set of facilities $\mathcal{F}_1$, the facility cost for opening $k \in \mathcal{F}_1$ is f_k (we can let $f_k = 0$ if k has been open in the current solution), and the revenue generated by assigning $j \in D$ to $k \in \mathcal{F}_1$ is $d_{jk} = \max\{0, o_j - c_{jki}\}$. Again, we solve the Max-1-LFLP by using Algorithm MAX. Assume that client j is assigned to $\sigma_i(j)$ by Algorithm MAX. If the total profit of the solution for the Max-1-LFLP is greater than f_i, then the total cost of the current solution for the 2-LFLP can be reduced by opening facility i, and for each client j, if $d_{j\sigma_i(j)} > 0$ then open $\sigma_i(j)$ and re-connect j to the path $(\sigma_i(j), i)$. Such a step is called an 'add operation'. Repeat this procedure until the solution can not be improved by any 'add operation'

Consider any feasible solution, say OPT, for the 2-LFLP. Assume its total connection cost and facility cost are C^{opt} and F^{opt}. Again, w.l.o.g., we can assume that the solution OPT forms a forest. Consider one of the trees rooted at $i \in \mathcal{F}_2$. Let S_i and D_i be defined as those in the last section. Now we are ready to prove the following Lemma, whose counterpart for the 1-LFLP is well-known in this area.

Lemma 4. *If no more 'add operation' can improve the current solution, then* $C \leq C^{opt} + \frac{e}{e-1}F^{opt}$.

Proof. Let $OPT(\mathcal{F}_i)$ be the subset of facilities in $\mathcal{F}_i$ that are open in the feasible solution OPT, for $i = 1, 2$. In the local improvement procedure, assume that we are considering a candidate facility $i \in OPT(\mathcal{F}_2)$. Let the connection cost of client j be, o_j in the current solution, and θ_j in OPT.

As in the proof of Lemma 3, there exists a feasible solution for the Max-1-LFLP with total profit at least

$$\sum_{j\in D_i} \max\{o_j - \theta_j, 0\} - \sum_{k\in S_i} f_k.$$

Therefore, Algorithm MAX can find a solution with total profit at least

$$\left(1 - \frac{1}{e}\right) \sum_{j\in D_i} \max\left\{o_j - \theta_j, 0\right\} - \sum_{k\in S_i} f_k \geq \left(1 - \frac{1}{e}\right) \sum_{j\in D_i} \left(o_j - \theta_j\right) - \sum_{k\in S_i} f_k.$$

However, since no 'add operation' can improve the current solution, we must have

$$f_i \geq \left(1 - \frac{1}{e}\right) \sum_{j\in D_i} \left(o_j - \theta_j\right) - \sum_{k\in S_i} f_k.$$

Note that for $i \neq l$, $D_i \cap D_l = \emptyset$, $\cup_{i\in OPT(\mathcal{F}_2)} D_i = D$ and $\cup_{i\in OPT(\mathcal{F}_2)} F_1^i = OPT(\mathcal{F}_1)$. Then we add the above inequality over $i \in OPT(\mathcal{F}_2)$, which completes the proof. □

The above lemma leads to the following conclusion immediately:

Lemma 5. *If there is an* (a, b)*-approximation algorithm for the* 2*-LFLP, then we can get an approximation algorithm with performance guarantee* $\left(a + \frac{e}{e-1}(\Delta - 1), 1 + \frac{b-1}{\Delta}\right)$ *for any* $\Delta \geq 1$.

Proof. Assume that there is an (a, b)-approximation algorithm for 2-LFLP, then by scaling the facility cost of the original instance by a factor of $\Delta \geq 1$, we can find a solution such that

$$\Delta F + C \leq a\Delta F^{OPT} + bC^{OPT}.$$

Then we apply the local improvement procedure on this solution, until the cost (on the scaled instance) can not be reduced anymore. Then we must have, by Lemma 4,

$$C \leq C^{OPT} + \frac{e}{e-1}\Delta F^{OPT}.$$

Combining these two inequalities, we must have

$$F + C \le \left(a + \frac{e}{e-1}(\Delta - 1)\right) F^{OPT} + \left(1 + \frac{b-1}{\Delta}\right) C^{OPT}.$$

□

Theorem 3. *For any given $\epsilon > 0$, if there is an (a, b)-approximation algorithm for the 2-LFLP, then we can get an approximation algorithm with performance guarantee*

$$\left(a + \frac{e}{e-1}\ln(\Delta) + \epsilon, 1 + \frac{b-1}{\Delta}\right)$$

for any $\Delta \ge 1$.

Proof. Assume that we have an (a, b)-approximation algorithm for 2-LFLP. Let $\delta \ge 1$. We prove that for any integer $p \ge 1$, there exists an approximation algorithm for 2-LFLP with performance guarantee $\left(a + \frac{e}{e-1}p(\delta - 1), 1 + \frac{b-1}{\delta^p}\right)$.

We prove the claim by induction on p. The case $p = 1$ is trivial by Lemma 5.

Assume that the claim is correct for $p-1$. Then we have an $\left(a + \frac{e(p-1)(\delta-1)}{e-1}, 1 + \frac{b-1}{\delta^{p-1}}\right)$-approximation algorithm for 2-LFLP. We apply Lemma 5 again, then we get an approximation algorithm with performance guarantee

$$\left(a + \frac{e(p-1)(\delta-1)}{e-1} + \frac{e(\delta-1)}{e-1}, 1 + \frac{1 + \frac{b-1}{\delta^{p-1}} - 1}{\delta}\right),$$

which is exactly what we need.

Thus we have proved the claim for any integer p. Now for any $\Delta \ge 1$, let $\delta = \Delta^{\frac{1}{p}}$ for some large integer p. We have thus proved that any (a, b)-approximation algorithm implies an $\left(a + \frac{e}{e-1}p\left(\Delta^{\frac{1}{p}} - 1\right), 1 + \frac{b-1}{\Delta}\right)$-approximation algorithm. Note that

$$p\left(\Delta^{\frac{1}{p}} - 1\right) \to \ln\Delta \quad \text{when} \quad p \to \infty.$$

Thus, for any $\epsilon > 0$, there exists a constant p such that

$$p\left(\Delta^{\frac{1}{p}} - 1\right) \le \ln\Delta + \epsilon.$$

This completes the proof of the Theorem. □

5. Variants of the 2-LFLP

In this section, we present improved results on approximating several variants of the 2-LFLP.

5.1. The non-metric 2-LFLP

We show that Algorithm QG has a performance guarantee of $O(\ln(|\mathcal{D}|))$ for the non-metric 2-LFLP. The analysis is the same as that of the metric 2-LFLP, except that in Lemma 3, one of the inequalities of (5) does not necessarily hold. In fact, we can prove that

Lemma 6. *Let $f = \frac{e}{e-1}\left(f_i + \sum_{k\in S_i} f_k\right)$, then the system of inequalities (6) holds:*

$$\begin{aligned} &\forall\, 1 \le j < k:\ \alpha_j \le \alpha_{j+1} \\ &\forall\, 1 \le j \le k:\ \sum_{l=j}^{k} \max(\alpha_j/(1+\epsilon) - m_l, 0) \le f \\ &\forall\, 1 \le j \le k:\ \alpha_j, m_j, f \ge 0. \end{aligned} \tag{6}$$

This leads to the following theorem.

Theorem 4. *The non-metric 2-LFLP can be approximated by a factor of $O(\ln(|\mathcal{D}|))$ in polynomial time.*

Proof. The inequality system 6 implies that, for each $j : 1 \le j \le k$:

$$\alpha_j \le \frac{1}{k-j+1}\left(\frac{e}{e-1}\left(f_i + \sum_{k\in S_i} f_k\right) + \sum_{l=1}^{k} m_l\right).$$

It follows that

$$\sum_{j=1}^{k} \alpha_j \le H_k \left(\frac{e}{e-1}\left(f_i + \sum_{k\in S_i} f_k\right) + \sum_{l=1}^{k} m_l\right),$$

where $H_k = \sum_{i=1}^{k} \frac{1}{i} \le \ln(k)$. This completes the proof of the theorem. □

To the best of our knowledge, no approximation algorithm for the non-metric 2-LFLP is known in the literature. Hochbaum [23] has shown that the non-metric 1-LFLP can be approximated by a factor of $\ln(|\mathcal{D}|)$, and by a result of Feige [19], it is the best possible unless P=NP. Therefore, our ratio for the non-metric 2-LFLP is the best possible up to a constant.

5.2. The two-level concentrator location problem

In the two-level concentrator location problem (2-LCLP), we have the same input as that of the 2-LFLP. However, each client must be served by a first level open facility only, and each of the first level open facilities must be served by a second level open facility. To be precise, we are asked to choose subset $\emptyset \ne V_t \subseteq \mathcal{F}_t$ to open, for $t = 1, 2$ such that

$$\sum_{j\in D} \min_{k\in V_1} c_{jk} + \sum_{k\in S_1} \min_{i\in V_2} c_{ki} + \sum_{t=1}^{2} \sum_{i_t\in V_t} f_{i_t}$$

is minimized. Here, we assume that the connection cost satisfy the triangle inequality.

As that for the 2-LFLP, we can assume that the open paths of an optimal solution of the 2-LCLP form a forest. We consider any tree of the forest with its root $i \in \mathcal{F}_2$. Again, we denote the leaves of the tree by S_i that is a subset of $\mathcal{F}_1$. And D_i is the set of clients that are assigned to the tree rooted at i in the optimal solution. Therefore, the total cost (of the optimal solution) associated with this tree is

$$f_i + \sum_{k \in S_i} (f_k + c_{ki}) + \sum_{j \in D_i} \min_{k \in S_i} c_{jk}.$$

If we could prove that, for each $i \in \mathcal{F}_2$,

$$\sum_{j \in D_i} \alpha_j \leq R_f \cdot \left(f_i + \sum_{k \in S_i} (f_k + c_{ki}) \right) + R_c \cdot \sum_{j \in D_i} \min_{k \in S_i} c_{jk},$$

then Algorithm QG must be a (R_f, R_c)-approximation algorithm for the 2-LCLP.

We can apply Algorithm QG to the 2-LCLP as well. The only change we should make is the construction of the instances of the Max-1-LFLP. In particular, when considering a facility $i \in \mathcal{F}_2$, the instance of the Max-1-LFLP is constructed as follows: the set of clients is $\mathcal{D}$, the set of facilities is $\mathcal{F}_1$, the facility cost for opening facility $k \in \mathcal{F}_1$ is $f_k + c_{ki}$, and the revenue generated by assigning client j to facility k is $\max\{0, B_j - c_{jk}\}$, where $B_j = c_{jk'}$ if j is currently connected to a facility $k' \in \mathcal{F}_1$, otherwise $B_j = \alpha_j$. Then following the same analysis as that for 2-LFLP, we can show that the 2-LCLP can be approximated by a factor of $1.77(1+\epsilon)^2$ in polynomial time for any constant $\epsilon > 0$. Levin [26] claimed a constant factor approximation algorithm for the k-LCLP when k is a constant. The approximation ratio of his algorithm is exponential on k. Our result significantly improves the ratio for $k = 2$.

5.3. The 3- *and the* 4-*LFLP*

In [5], improved algorithms for the k-LFLP have been proposed by reducing it to the 1-LFLP. In fact, in order to get improved ratios for the k-LFLP, one needs to combine two reduction methods: the parameterized path reduction and the recursive reduction. For details of the reductions, we invite the author to refer to [5].

Since we have a strong approximation for the 2-LFLP, we can further refine their reduction. We omit the proofs of Lemma 7 and Lemma 8, since complete proofs would essentially repeat all the arguments of [5]. For the 3-LFLP, we use exactly the same reductions as those in [5]. However, when we apply the recursive reduction to the 3-LFLP, we need to solve an instance of the 2-LFLP. In [5], the instance of the 2-LFLP is further reduced to two instances of the 1-LFLP. Instead of doing this, we now can solve the instance of the 2-LFLP directly by using Algorithm QG. This leads to the following lemma.

Lemma 7. *Assume that the* 1- *and the* 2-*LFLP can be approximated by factors of* (a, b) *and* (α, β)*, respectively, then the* 3-*LFLP can be approximated by factors of* $\left(\max\left\{a, \frac{a+\alpha}{2}\right\}, \frac{3b+\beta}{2}\right)$.

Therefore, we can obtain a better approximation ratio for the 3-LFLP. The previously best known ratio is 2.85.

Theorem 5. *The* 3*-LFLP can be approximated by a factor of* 2.51.

Proof. By Theorem 2 and Theorem 3, and by letting $\Delta = 1.262196$, we know that the 2-LFLP can be approximated by a factor of (α, β) such that

$$\alpha = e/(e-1)\,(1.118 + \ln(1.262196))\,, \quad \beta(\Delta) = 1 + 0.77/1.262196.$$

By Lemma 1, and by letting $\delta = 5.991324$, we know that the 1-LFLP can be approximated by a factor of $(a, b) = (1.104 + \ln(5.991324), 1 + 0.7805/5.991324)$. It follows from Lemma 7 that the 3-LFLP can be approximated by a factor of 2.51. □

For the 4-LFLP, we modify the reductions of [5] in the following way. In the parameterized path reduction, for any instance of the 4-LFLP, we reduce it to an instance of the 2-LFLP. In the recursive path reduction, we reduce any instance of the 4-LFLP to two instances of the 2-LFLP (in [5], it will be reduced to one instance of the 1-LFLP, and one instance of the 3-LFLP). We can prove the following lemma.

Lemma 8. *Assume that the* 2*-LFLP can be approximated by a factor of* (α, β)*, then the* 4*-LFLP can be approximated by factors of* $(\alpha, 2\beta)$.

Theorem 6. *The* 4*-LFLP can be approximated by a factor of* 2.81.

Proof. By Theorem 2 and Theorem 3, we know that the 4-LFLP can be approximated by a factor of (α, β) such that

$$\alpha(\Delta) = e/(e-1)(1.118 + \ln\Delta), \beta(\Delta) = 1 + 0.77/\Delta$$

for some $\Delta \geq 1$. By letting $\Delta = 1.92$, we have $\alpha \leq 2.803$ and $\beta \leq 1.402$. Therefore, by Lemma 8, we get a 2.81-approximation algorithm for the 4-LFLP. □

We remark that no approximation ratio better than 3 was known for the k-LFLP for any $k \geq 4$.

5.4. The 2*-LFLP with soft capacities*

This problem is the same as the 2-LFLP except that the facility cost for opening facility i is a function of the number of clients it serves. In particular, if i serves k clients, then the facility cost of i is $f_i \left\lceil \frac{k}{u_i} \right\rceil$ where f_i and u_i are given. For each facility i, there is an upper bound u_i on the number of clients it can serve. It has been shown in [5] (Theorem 2, page 154) that the 2-LFLP can be approximated by a factor of (1.104, 3.56). By Theorem 3 and by letting $\Delta = 1.281$, we obtain a (1.5, 3)-approximation algorithm for the 2-LFLP. We follow the approach of [31] and show that any (α, β)-approximation algorithm for the 2-LFLP implies a $(2\alpha, 2\beta)$-approximation algorithm for the 2-LFLP with soft capacities. Therefore, we can get a 3-approximation algorithm for the 2-LFLP with soft capacities.

6. Concluding Remarks

We remark that one may formulate the 2-LFLP as an integer program with exponentially many variables. Then our algorithm can be viewed as a dual-ascent algorithm that is similar to [28]. However, for the 2-LFLP, the dual of the linear programming relaxation has exponentially many constraints, and it is NP-hard to find an efficient separation oracle. Our algorithm for the Max-1-LFLP gives an approximate separation oracle for the dual problem. We notice that Jain et al. [29] and Carr et al. [9] have used approximate separation oracles, combined with ellipsoid method, to solve linear programs with exponentially many constraints.

Several questions remain open. First of all, although our bound does not depend on the LP relaxation for the 2-LFLP [37], it would be of interest to know the integrality gap of the LP relaxation. A related question is on the lower bound for the optimal performance guarantee for the 2-LFLP, i.e., whether we can improve it from 1.463, which is the known lower bound for the 1-LFLP.

Although in many applications of the multi-level problem, the number of levels is small, it is certainly of theoretical interest to study the approximability of the k-LFLP for general k. Can our technique be extended to the k-LFLP for $k \geq 2$? This requires a good approximation for the maximization version of the $(k-1)$-LFLP, which has been studied in [6] and [39]. But their results are not in the form that can be applied within our framework. Finally, our results, if combined with that of Aardal, Chudak, and Shmoys [2] strongly suggest that it is possible to develop a polynomial time algorithm for the k-LFLP whose performance guarantee depends on k, i.e., the performance guarantee is strictly less than 3 for any k, and converges to 3 when k goes to infinity.

Acknowledgements. The author would like to thank Yinyu Ye for his support and his feedbacks on preliminary drafts of this paper. The author also thanks Alexander Ageev, Asaf Levin, Pete Veinott, Dachuan Xu, SODA 2004 program committee, and anonymous referees for their helpful comments. In particular, the proof of Theorem 1, which is much simpler than the original proof, was suggested by one of the referees. This research was supported in part by NSF grant DMI-0231600 through Yinyu Ye.

References

1. Aardal, K., Labbe, M., Leung, J., Queyranne, M.: On the two-level uncapacitated facility location problem. INFORMS Journal on Computing **8**, 289–301 (1996)
2. Aardal, K., Chudak, F.A., Shmoys, D.B.: A 3-approximation algorithm for the k-level uncapacitated facility location problem. Information Processing Letters **72**, 161–167 (1999)
3. Ageev, A.: Improved approximation algorithms for multilevel facility location problems. Oper. Res. Letters **30**, 327–332 (2002)
4. Ageev, A., Sviridenko, M.: An 0.828-approximation algorithm for uncapacitated facility location problem. Discrete Applied Mathematics **93**, 289–296, (1999)
5. Ageev, A., Ye, Y., Zhang, J.: Improved Combinatorial Apporixmation Algorithms for the k-Level Facility Location Problem. In: Proceedings of The 30th International Colloquium on Automata, Languages and Programming (ICALP), LNCS 2719, 2003, pp. 145–156
6. Bumb, A.: An approximation algorithm for the maximization version of the two level uncapacitated facility location problem. Operations Research Letters **29**(4), 155–161 (2001)
7. Bumb, A.F., Kern, W.: A simple dual ascent algorithm for the multilevel facility location problem. In: 4th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2001), LNCS 2129, 2001, pp. 55–62
8. Barros, A.I., Labbe, M.: A general model for the uncapacitated facility and depot location problem. Location Science **2**, 173–191 (1994)

9. Carr, R.D., Fleischer, L., Leung, V.J., Phillips, C.A.: Strengthening integrality gaps for capacitated network design and covering problems. In: Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2000, pp. 106–115
10. Chardaire, P.: Facility Location Optimization and Cooperative Games. Ph.d. Thesis, School of Information Systems, University of East-Anglia, Norwich NR4 7TJ, UK, 1998
11. Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation Algorithms for Directed Steiner Problems. Journal of Algorithms **33**, 73–91 (1999)
12. Charikar, M., Guha, S.: Improved combinatorial algorithms for facility location problems. SIAM Journal on Computing **34**(4), 803–824 (2005)
13. Chekuri, C., Even, G., Guy Kortsarz: A combinatorial approximation algorithm for the group Steiner problem. submitted to *Discrete Applied Mathematics*
14. Cornuéjols, G., Fisher, M.L., Nemhauser, G.L.: Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. Mngt Sci. **23**, 789–810 (1977)
15. Cornuéjols, G., Nemhauser, G.L., Wolsey, L.A.: The uncapacitated facility location problem. In: P. Mirchandani and R. Francis, (eds.) Discrete Location Theory, Wiley, New York, 1990, pp. 119–171
16. Chudak, F.A., Shmoys, D.B.: Improved approximation algorithms for the uncapacitated facility location problem. SIAM Journal on Computing **33**, 1–25 (2003)
17. Edwards, N.: Approximation algorithms for the multi-level facility location problem. Ph.D. Thesis, School of Operations Research and Industrial Engineering, Cornell University, 2001
18. Erdös, P., Selfridge, J.L.: On a combinatorial game. J. Combinatorial Theory, Ser. A **14**, 298–301 (1973)
19. Feige, U.: A threshold of $\ln n$ for approximating set cover. Journal of the ACM **45**, 634–652 (1998)
20. Gao, J.J., Robinson, E.P. Jr.: A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. Naval Research Logistics **839**, 191–212 (1992)
21. Gao, J.J., Robinson, E.P. Jr.: Uncapacitated facility location: General solution procedure and computational experience. European Journal of Operational Research **76**, 410–427 (1994)
22. Guha, S., Khuller, S.: Greedy strikes back: improved facility location algorithms. Journal of Algorithms **31**, 228–248 (1999)
23. Hochbaum, D.S.: Heuristics for the fixed cost median problem. Mathematical Programming **22**(2), 148–162 (1982)
24. Kaufman, L., Eede, M.V., Hansen, P.: A plant and warehouse location problem. Operations Research Quarterly **28**, 547–554 (1977)
25. Labbe, M.: The multi-level uncapacitated facility location problem is not submodular. European Journal of Operational Research **72**, 607–609 (1996)
26. Levin, A.: Personal Communication, 2003
27. Guha, S., Meyerson, A., Munagala, K.: Hierarchical placement and network design problems. In: Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS), 2000, pp. 603–612
28. Jain, K., Mahdian, M., Saberi, A.: A new greedy approach for facility location problems. In: Proceedings of the 34th ACM Symposium on Theory of Computing (STOC), 2002, pp. 731–740
29. Jain, K., Mahdian, M., Salavatipour, M.R.: Packing Steiner trees. In: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2003, pp. 266–274
30. Mahdian, M., Ye, Y., Zhang, J.: Improved approximation algorithms for metric facility location problems. In: 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2002), Lecture Notes in Computer Science, 2462, 2002, pp. 229–242
31. Mahdian, M., Ye, Y., Zhang, J.: A 2-approximation algorithm for the soft-capacitated facility location problem. In: 6th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2003), Lecture Notes in Computer Science, 2764, 2003, pp. 129–140
32. Meyerson, A., Munagala, K., Plotkin, S.: Cost-distance: two-metric network design. In: Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS), 2000, pp. 624–630
33. Ro, H., Tcha, D.: A branch-and-bound algorithm for the two-level uncapacitated facility location problem with some side constraints. European Journal of Operational Research **18**, 349–358 (1984)
34. Robinson, E.P. Jr., Gao, L.: A new formulation and linear programming based optimization procedure for the two-echelon uncapacitated facility location problem. Annals of the Society of Logistics Engineers **2**, 39–59
35. Robinson, E.P. Jr., Gao, L., Muggenborg, S.D.: Designing an integrated distribution system at DowBrands Inc. Interfaces **23**, 107–117 (1993)
36. Shmoys, D.B.: Approximation algorithms for facility location problems. In: 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX), LNCS 1913, 2000, pp. 27–33
37. Shmoys, D.B., Tardos, E., Aardal, K.I.: Approximation algorithms for facility location problems. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing 29th STOC, 1997, pp. 265–274
38. Tcha, D., Lee, B.: A branch-and-bound algorithm for the multi-level uncapacitated facility location problem. European Journal of Operational Research **18**, 35–43 (1984)
39. Zhang, J., Ye, Y.: A Note on the Maximization Version of the Multi-level Facility Location Problem. Operations Research Letters **30**(5), 333–335 (2002)