

Lower-Bounded Facility Location

Zoya Svitkina*

Abstract

We study the lower-bounded facility location problem, which generalizes the classical uncapacitated facility location problem in that it comes with lower bound constraints for the number of clients assigned to a facility in the case that this facility is opened. This problem was introduced independently in the papers by Karger and Minkoff [12] and by Guha, Meyerson, and Munagala [7], both of which give bicriteria approximation algorithms for it. These bicriteria algorithms come within a constant factor of the optimal solution cost, but they also violate the lower bound constraints by a constant factor. Our result in this paper is the first true approximation algorithm for the lower-bounded facility location problem, which respects the lower bound constraints and achieves a constant approximation ratio for the objective function. The main technical idea for the design of the algorithm is a reduction to the capacitated facility location problem, which has known constant-factor approximation algorithms.

1 Introduction

In the uncapacitated facility location (FL) problem, we are given a set of clients and a set of facilities, as well as a metric specifying the distances between the clients and the facilities. The goal is to choose a subset of facilities to open, and to assign each client to an open facility, in such a way as to minimize the sum of the facility opening costs and the connection costs. A facility opening cost for each facility is specified as part of the input, and it has to be paid if the facility is opened. A connection cost for each client is its distance to the facility to which it is assigned. The facility location problem has been studied extensively. It is NP-hard, but a number of good constant-factor approximation algorithms are known for it, using several techniques such as linear program rounding, primal-dual method and local search. Facility location has been used to model practical scenarios, such as the location of stores or warehouses in a geographical area, or of servers on a network. It is also used as a subroutine to solve more complex optimization problems.

The lower-bounded facility location (LBFL) problem that we consider in this paper is an extension of the uncapacitated facility location, as it includes an extra set of constraints. In particular, in addition to all the elements of a regular facility location instance, an instance of the lower-bounded version specifies a lower bound B , which is the minimum number of clients that can be assigned to a facility if it is opened. Obviously, if B is equal to zero, then the problem reduces to the original facility location. The LBFL problem was introduced simultaneously and independently by Karger and Minkoff [12], who use it as a subroutine for solving the *maybecast* network design problem, and by Guha, Meyerson and Munagala [7], who use it as a subroutine for solving the *access network design* problem. Both papers propose bicriteria approximation algorithms for LBFL, which violate both the lower bound constraints and the optimality of the objective function by constant factors, but are sufficient for their purposes.

As demonstrated by the algorithms of [12] and [7], as well as [8], LBFL can be a useful subroutine for solving various network design problems. Undoubtedly, the problems presented in those papers are not the only ones for which the solution of LBFL would be useful. In addition, the LBFL problem formulation has direct applications. For example, Lim, Wang, and Xu [14] present a transportation problem faced by a real-world company that has to decide on allocation of cargo from customers (‘clients’) to carriers (‘facilities’), who then ship it overseas. There is a transportation cost per unit demand assigned from each customer to each carrier, which can be modeled by the connection cost. But the main difficulty arises from the fact that there is a regulation enforcing a “minimum quantity commitment”, i.e. a rule that the total amount of cargo delivered by each carrier, if any, must be at least a certain minimum quantity. So the problem becomes exactly LBFL, but without facility costs (which seems to be as hard as the general LBFL). Other example applications of LBFL include the location of stores, with the requirement that each individual store serve a given minimum number of customers to remain profitable [7], and a clustering problem in which each cluster has to be at least a certain size, while the average distance of data points to cluster centers is minimized [12].

*Department of Computer Science, Dartmouth College, Hanover, NH. Supported by NSF ITR grant CCR-0325453.

Related work There has been much work on designing approximation algorithms for the uncapacitated facility location problem. The first constant-factor approximation algorithm was proposed by Shmoys, Tardos, and Aardal [19], and is based on linear program rounding. Subsequently, other constant-factor approximation algorithms were designed, based on various techniques, including the primal-dual method and local search (e.g. [1, 2, 4, 11, 13, 20]). Currently the best approximation guarantee is 1.52 [16].

Lower-bounded facility location problem was introduced by Guha, Meyerson, and Munagala [7], who call it the *load-balanced facility location* and use it for solving the *access network design* problem, which is a special case of the single-sink buy-at-bulk problem. Simultaneously, LBFL was also introduced by Karger and Minkoff [12], who call it the *r-gathering* problem and use it to solve the *maybecast* problem, which models network design under uncertainty about demands. Both papers present essentially the same bicriteria approximation algorithm for LBFL, which, for any given constant $\alpha \in [0, 1)$, finds a solution which assigns at least $\alpha \cdot B$ clients to each open facility (where B is the lower bound on the number of clients) and costs at most $\frac{1+\alpha}{1-\alpha} \rho \cdot OPT$, where ρ is the approximation ratio for the FL problem, which is used as a subroutine, and OPT is the cost of the optimal solution to LBFL that respects the lower bound constraints. Thus, this algorithm provides a trade-off between the cost of the solution and the amount by which the lower-bound constraints are violated, but it is unable to find a truly feasible solution with a non-trivial guarantee on the cost. The LBFL problem is also considered by Lim et al. [14], who formulate it as a mixed-integer program and solve it using a branch-and-cut scheme. They also analyze a greedy heuristic for LBFL without facility costs and show that it is a $2B$ -approximation.

An extension of the facility location problem which in some sense is the opposite of LBFL is the capacitated facility location (CFL) problem. In CFL, each facility has a capacity, which is the maximum number of clients that can be assigned to it. This problem is significantly harder than the uncapacitated version. For example, all known LP relaxations for it have unbounded integrality gaps. However, there are several known constant-factor approximation algorithms for CFL, all of which are based on the local search technique. Korupolu, Plaxton and Rajaraman [13] gave a constant-factor approximation for the special case of uniform capacities, which was later improved by Chudak and Williamson [5]. The first constant-factor algorithm for non-uniform capacities, providing a $(8.53 + \varepsilon)$ -approximation, was given by Pál, Tardos and Wexler [18]. Currently the best bound

is $3 + 2\sqrt{2} + \varepsilon \leq 5.83 + \varepsilon$ [21]. A variant of the capacitated problem is facility location with soft capacities, in which facilities can be opened multiple times for extra cost, thus serving more clients than their capacity. This version of the problem is generally easier to solve than CFL, as it does not suffer from large integrality gaps, and can be reduced to the regular FL problem. A number of constant-factor approximation algorithms have been proposed for it [1, 3, 10, 11, 17].

A formulation that generalizes CFL with either hard or soft capacities, as well as a number of other problems, is known as the universal facility location problem. In it, instead of capacities, each facility has a cost function which depends on the number of clients that are assigned to it. For example, CFL can be modeled by a cost function that starts out as constant, but then goes to infinity when the number of clients exceeds the capacity. This formulation was introduced by Hajiaghayi, Mahdian and Mirrokni [9], who focus on the special case of concave functions and give a constant approximation based on a reduction to the uncapacitated problem. Subsequently, Mahdian and Pál [15] gave an algorithm that works for arbitrary monotone non-decreasing facility cost functions. Their algorithm is an extension of the local search technique of [18] for CFL, and gives a $7.88 + \varepsilon$ approximation. It was later improved by Garg, Khandekar and Pandit [6], who achieve a $3 + 2\sqrt{2} + \varepsilon$ approximation ratio, bridging the gap between known guarantees for universal facility location and CFL.

Our results and techniques In this paper, we present the first constant-factor true approximation algorithm for the lower-bounded facility location problem, thus resolving an open question of Karger and Minkoff [12]. Our algorithm is a true approximation in the sense that the produced solution is feasible for the original problem, satisfying the lower-bound constraints exactly. This is in contrast to bicriteria algorithms, which violate these constraints by constant factors. Whether or not a bicriteria approximation algorithm is an acceptable solution depends on the specific application. For example, in the contexts in which LBFL was originally introduced [7, 12], the bicriteria algorithms are sufficient for their purposes, and their violation of the constraints does not present major difficulties. However, in other cases, either in real-world applications or in reductions for other problems, a true approximation for the problem may be needed. For example, in the transportation application mentioned above, a bicriteria solution would not be satisfactory.

The main technical idea that we use for solving LBFL is to create an instance of the capacitated facility location problem by reversing the roles played by the

clients and the facilities. To give a rough description lacking many details, we can say that a group of clients at a given location becomes a facility whose capacity is the number of those clients. Conversely, a facility that has not yet been filled to the bound B becomes a client whose demand is the number of “slots” that still have to be filled in order for this facility to reach B . Then the task becomes to make an assignment which would use the clients to fill the “slots” in such a way that each open facility has at least B clients assigned to it. We use a CFL subroutine to make such an assignment, taking advantage of the known constant-factor approximation algorithms for it. Our actual algorithm for LBFL also involves a pre-processing step, in which we compute a bicriteria solution to our input instance, as well as a post-processing step, in which we assign some remaining left-over clients.

Overview Our algorithm consists of three main stages: first, we find a bicriteria-approximate solution and use it to transform the instance, taking care that the value of the optimal solution does not increase too much; then we use this modified instance to define a CFL problem, and solve it using one of the known algorithms; finally, based on the solution to the CFL instance, we transfer clients between facilities in a way that transforms the bicriteria solution into an approximate solution that does not violate the lower bound constraints. In the following sections, we begin with the formal problem definition and a review of the bicriteria algorithm in Section 2; then Sections 3, 4 and 5 describe the three stages of the algorithm respectively.

2 Problem definition and bicriteria algorithm

We begin with a precise statement of the problem.

DEFINITION 2.1. *An instance \mathcal{I} of the lower-bounded facility location problem consists of a set of clients \mathcal{D} , a set of facilities \mathcal{F} , a non-negative facility cost $f(i)$ for each facility $i \in \mathcal{F}$, a distance metric $c(i, j)$ on the set $\mathcal{D} \cup \mathcal{F}$, and a bound B . A feasible solution consists of a subset $\mathcal{O} \subseteq \mathcal{F}$ of facilities to open, and an assignment of each client to an open facility, so that each open facility has at least B clients assigned to it. For a given solution, we use $j \rightarrow i$ to denote the fact that client $j \in \mathcal{D}$ is assigned to facility $i \in \mathcal{F}$, and $i(j)$ to denote the facility to which $j \in \mathcal{D}$ is assigned. The objective then is to minimize*

$$\sum_{i \in \mathcal{O}} f(i) + \sum_j c(j, i(j)),$$

subject to $|\{j : j \rightarrow i\}| \geq B$ for all $i \in \mathcal{O}$.

Let $\text{OPT}(\mathcal{I})$, or just OPT , denote the value of the optimal solution to \mathcal{I} , with $C^* = \sum_j c(j, i(j))$ being its

connection cost and $F^* = \sum_{i \in \mathcal{O}} f(i)$ being its facility cost. Let $j \rightarrow^* i$ and $i^*(j)$ represent the assignments made by the optimal solution.

Our algorithm for LBFL uses the bicriteria approximation algorithm of [7, 12] as a first step, as described in more detail in the next section. Here, for the sake of completeness, let us review this algorithm and its analysis. The algorithm takes a parameter $\alpha \in [0, 1)$ and returns a solution which assigns at least αB clients to each open facility.

For each facility $i \in \mathcal{F}$, let $\mathcal{D}(i) \subseteq \mathcal{D}$ be the set of the closest B clients to i . We now construct an instance \mathcal{I}' of the FL problem by dropping the lower bounds from \mathcal{I} and setting facility costs to

$$f'(i) = f(i) + \lambda \sum_{j \in \mathcal{D}(i)} c(i, j).$$

Here $\lambda = \frac{2\alpha}{1-\alpha}$ is just a constant used for scaling. The idea behind the term $\sum_{j \in \mathcal{D}(i)} c(i, j)$ is that if a facility i is opened in a solution to LBFL, then, since it serves at least B clients, the connection cost of its clients will be at least this much. Once the instance \mathcal{I}' is constructed, we solve it using a ρ -approximation algorithm, ensuring that each client is assigned to its nearest open facility (this only improves the objective). Call the resulting solution \mathcal{S}' .

The second step of the bicriteria algorithm is to perform a reassignment of clients from open facilities that are serving less than αB clients in \mathcal{S}' . For each such facility i , in arbitrary order, and each client j assigned to it, do the following: find the nearest to j open facility $i' \neq i$ and reassign j from i to i' . When all clients from facility i are reassigned, close i . Note that the invariant that each client is assigned to its nearest open facility is maintained. Clearly, at the end of this procedure, each open facility is serving at least αB clients.

To analyze the algorithm, we make the following observations.

LEMMA 2.1. $\text{OPT}(\mathcal{I}') \leq (\lambda + 1)\text{OPT}(\mathcal{I})$.

Proof. Suppose the optimal solution to the LBFL instance \mathcal{I} opens a set of facilities \mathcal{O} , has facility cost F^* and connection cost C^* . This same solution is feasible for the FL instance \mathcal{I}' . Its connection cost for \mathcal{I}' is the same as it is for \mathcal{I} , C^* . Its facility cost for \mathcal{I}' is

$$\sum_{i \in \mathcal{O}} f'(i) = \sum_{i \in \mathcal{O}} \left[f(i) + \lambda \sum_{j \in \mathcal{D}(i)} c(i, j) \right] \leq F^* + \lambda C^*.$$

So the overall cost of this solution, which serves as an upper bound on $\text{OPT}(\mathcal{I}')$, is at most $F^* + (\lambda + 1)C^* \leq (\lambda + 1)\text{OPT}$. \square

Since the FL instance was solved using a ρ -approximation, we get the following corollary.

COROLLARY 2.1. *The cost of the solution \mathcal{S}' is at most $(\lambda + 1)\rho \cdot \text{OPT}(\mathcal{I})$.*

Now we analyze the additional cost incurred by the second step of the algorithm.

LEMMA 2.2. *The additional connection cost incurred by transferring clients from any facility i in the second step of the algorithm is at most $f'(i)$, the facility cost of i in \mathcal{I}' .*

Proof. To bound this cost, we observe that for any facility i with less than αB clients assigned to it by \mathcal{S}' , there must be at least $(1 - \alpha)B$ clients that are included in the set $\mathcal{D}(i)$ but are not assigned to i . Since the total distance of all clients in $\mathcal{D}(i)$ to i is $\sum_{j \in \mathcal{D}(i)} c(i, j)$, the average distance of those $(1 - \alpha)B$ clients is at most $\frac{\sum_{j \in \mathcal{D}(i)} c(i, j)}{(1 - \alpha)B}$, and so is the minimum distance between i and one of these clients, say j' (see Figure 1). Because

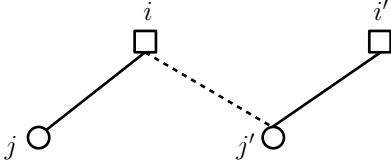


Figure 1: Use of triangle inequality in Lemma 2.2.

j' is assigned to its nearest open facility $i' \neq i$, we have that $c(j', i') \leq c(j', i)$. This means that there must be another open facility (namely i') at a distance of at most $2 \cdot \frac{\sum_{j \in \mathcal{D}(i)} c(i, j)}{(1 - \alpha)B}$ from i (by using the triangle inequality on the distances from i to j' and from j' to i'). So when a client j is reassigned from facility i to its nearest open facility, the increase in distance for it is at most $2 \cdot \frac{\sum_{j \in \mathcal{D}(i)} c(i, j)}{(1 - \alpha)B}$. As there are at most αB such clients, the total additional connection cost that we pay for reassigning them from i is at most

$$2 \frac{\alpha B}{(1 - \alpha)B} \sum_{j \in \mathcal{D}(i)} c(i, j) = \lambda \sum_{j \in \mathcal{D}(i)} c(i, j) \leq f'(i).$$

□

Overall, we get the following approximation guarantee.

THEOREM 2.1. *The solution found by the bicriteria algorithm for the LBFL instance \mathcal{I} has cost at most $\frac{1 + \alpha}{1 - \alpha} \rho \cdot \text{OPT}(\mathcal{I})$.*

Proof. The cost of the final solution consists of the following parts: the original connection cost, which is equal to the connection cost of \mathcal{S}' ; the facility cost of facilities that remain open, which is at most the facility cost of these facilities in \mathcal{S}' ; and the additional connection cost for reassignments, which is at most the facility cost of the facilities that were closed. So the total cost is at most that of \mathcal{S}' , and substituting the definition of λ into Corollary 2.1, we get the result. □

3 Transforming the instance

In order to apply the main step of our algorithm, which uses a CFL subroutine, we simplify the problem in a few ways, ensuring that the new instance has some useful properties. In particular, it does not have facility costs, and has clients clustered in relatively large groups (a constant fraction of B) at each location. To do this, we employ the bicriteria approximation algorithm described in the previous section. We consider two modified instances of the LBFL problem, instance \mathcal{I}_1 obtained by modifying the original problem according to the bicriteria solution, and instance \mathcal{I}_2 obtained by further modifying \mathcal{I}_1 (see Figure 2). In this section we define these instances and bound the values of their optimal solutions in terms of the optimum for the original problem.

The bicriteria algorithm is applied to the original problem instance \mathcal{I} , with a parameter $\alpha > \frac{1}{2}$ to be specified later. Let $j \rightarrow^b i$ and $i^b(j)$ denote the assignments made by the obtained solution. Also, let C^b and F^b denote its connection and facility costs, respectively. We now define the first modified instance, \mathcal{I}_1 .

DEFINITION 3.1. *Let \mathcal{I}_1 be an instance of LBFL, whose elements \mathcal{D} , \mathcal{F} , and B are the same as in \mathcal{I} , but the metric of distances and the facility costs are different. The distances are modified as follows. Intuitively, every client is “moved” to the location of the facility to which it is assigned by the bicriteria solution. Formally, for any two clients $j, j' \in \mathcal{D}$ and two facilities $i, i' \in \mathcal{F}$, the distances become: $c_1(j, i) = c(i^b(j), i)$; $c_1(j, j') = c(i^b(j), i^b(j'))$; and the distance between facilities remains the same, $c_1(i, i') = c(i, i')$. The facility costs are modified so that all the facilities that are opened by the bicriteria solution become free, and the costs of others remain the same: $f_1(i) = 0$ if there exists $j \in \mathcal{D}$ such that $j \rightarrow^b i$, and $f_1(i) = f(i)$ otherwise.*

It's not hard to see that the new distances c_1 also form a metric. The cost of the optimal solution to \mathcal{I}_1 can be bounded as follows.

LEMMA 3.1. $\text{OPT}(\mathcal{I}_1) \leq \frac{1 + \alpha}{1 - \alpha} \cdot \rho \cdot \text{OPT} + \text{OPT}$.

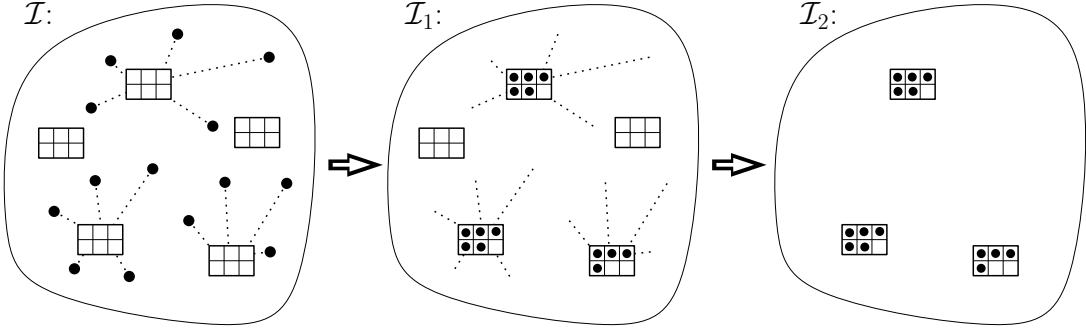


Figure 2: An example of defining the instances \mathcal{I}_1 and \mathcal{I}_2 . The black circles represent the clients, and the large rectangles represent the facilities, whose lower bound is $B = 6$. The dotted lines show the assignment of clients to facilities made by the bicriteria algorithm for the original instance, with $\alpha B = 4$.

Proof. One feasible solution to \mathcal{I}_1 is to assign each client j to its optimal facility $i^*(j)$. The facility costs of this solution are at most those in OPT , F^* , and the connection cost for a client j is $c_1(j, i^*(j)) = c(i^b(j), i^*(j)) \leq c(i^b(j), j) + c(j, i^*(j))$ by the triangle inequality. Intuitively, j can be first moved back to its original location, and then moved from there to its optimal facility. Summing the connection costs over all clients, we get that the connection cost of this solution is $\sum_{j \in \mathcal{D}} c_1(j, i^*(j)) \leq C^b + C^*$. Since $C^b \leq \frac{1+\alpha}{1-\alpha} \cdot \rho \cdot \text{OPT}$ by the guarantee of the bicriteria algorithm, we get the result. \square

The second transformation that we make is to produce a LBFL instance \mathcal{I}_2 out of instance \mathcal{I}_1 by removing the facilities which are not used by the bicriteria solution that we found (see Figure 2).

DEFINITION 3.2. Let \mathcal{I}_2 be the same as \mathcal{I}_1 , except for the set of facilities, which becomes

$$\mathcal{F}_2 = \{i \in \mathcal{F} : j \rightarrow^b i \text{ for some } j \in \mathcal{D}\}.$$

Next we bound the cost of the optimal solution to \mathcal{I}_2 in terms of $\text{OPT}(\mathcal{I}_1)$.

LEMMA 3.2. $\text{OPT}(\mathcal{I}_2) \leq 2 \cdot \text{OPT}(\mathcal{I}_1)$

Proof. Consider the optimal solution to \mathcal{I}_1 , and suppose it uses some facility $i \notin \mathcal{F}_2$. Then, instead, transfer all clients from i to its closest facility $i' \in \mathcal{F}_2$. This is a feasible solution, since i' now has at least B clients. The facility cost did not increase, because i' has cost 0 (by the definition of facility costs in \mathcal{I}_1). To bound the possible increase in connection costs, observe that in \mathcal{I}_1 , each client is co-located with (i.e., is at distance 0 from) some facility in \mathcal{F}_2 . Now for the facility $i \notin \mathcal{F}_2$, let j be the closest client assigned to i . It must therefore be

that $c_1(i, i') \leq c_1(i, j)$. As a result, the total increase in cost from transferring clients from i to i' is at most

$$\sum_{j' \rightarrow i} c_1(i, i') \leq \sum_{j' \rightarrow i} c_1(i, j) \leq \sum_{j' \rightarrow i} c_1(i, j'),$$

where the second inequality follows because j was defined as the closest client assigned to i . Since the additional connection cost incurred for transferring clients from facility i is at most their original connection cost, the overall connection cost at most doubles, implying the result of the lemma. \square

In the following sections we show how to obtain a constant-factor approximation to \mathcal{I}_2 . The next lemma summarizes its relation to the original problem.

LEMMA 3.3. Let \mathcal{S} be a solution to \mathcal{I} that makes the same assignments as a β -approximate solution to \mathcal{I}_2 . Then its cost is at most

$$\left[(2\beta + 1) \frac{1 + \alpha}{1 - \alpha} \cdot \rho + 2\beta \right] \cdot \text{OPT}(\mathcal{I}).$$

Proof. If a solution to \mathcal{I}_2 with total cost $\beta \cdot \text{OPT}(\mathcal{I}_2)$ assigns client j to facility i , it pays a connection cost of $c_1(j, i) = c(i^b(j), i)$. If \mathcal{S} makes the same assignment, then it has to pay a connection cost of $c(j, i)$, covering the distance from j 's original location. Since $c(j, i) \leq c(j, i^b(j)) + c(i^b(j), i)$, the total connection cost of \mathcal{S} for all clients is at most $C^b + \beta \cdot \text{OPT}(\mathcal{I}_2)$. As the solution to \mathcal{I}_2 uses only the facilities in \mathcal{F}_2 , the facility cost of \mathcal{S} is at most F^b . So the total cost of \mathcal{S} is bounded by

$$\begin{aligned} C^b + F^b + \beta \cdot \text{OPT}(\mathcal{I}_2) &\leq \frac{1 + \alpha}{1 - \alpha} \rho \cdot \text{OPT} + 2\beta \cdot \text{OPT}(\mathcal{I}_1) \\ &\leq \frac{1 + \alpha}{1 - \alpha} \cdot \rho \cdot \text{OPT} + 2\beta \cdot \left(\frac{1 + \alpha}{1 - \alpha} \cdot \rho \cdot \text{OPT} + \text{OPT} \right) \\ &= \left[(2\beta + 1) \frac{1 + \alpha}{1 - \alpha} \cdot \rho + 2\beta \right] \cdot \text{OPT}, \end{aligned}$$

using Theorem 2.1 and Lemmas 3.2 and 3.1. \square

4 Reduction to capacitated facility location

At this point, we have an instance \mathcal{I}_2 of the LBFL problem which has special structure. It consists of a set of facilities, each of which with at least αB clients at distance 0 from it. Let us say that these clients, whose number is $n_i \geq \alpha B$, are *at* this facility i . The instance does not have facility costs, so its solution requires that the clients be somehow reassigned, possibly closing some of the facilities, so that the remaining facilities have at least B clients each, while minimizing the connection cost of the reassignments. Since with $\alpha > \frac{1}{2}$, the number of clients from any two facilities is sufficient to reach the bound of B , an initial idea of how to solve this problem might be to find some kind of a matching on the set of facilities. However, a simple example shows that this can be far from optimum. Consider a set of B facilities, each with $B - 1$ clients, located in a uniform metric space (with all distances equal to 1). Then the optimal solution is to close one of the facilities, reassigning one client from it to each of the other facilities, which costs $B - 1$ in connection cost. However, if the facilities are paired up by a matching, then the connection cost incurred is $\frac{B}{2}(B - 1)$.

The way we solve the special case of the LBFL problem presented by the instance \mathcal{I}_2 is by using a reduction to the capacitated facility location problem. The general idea is that the clients from those locations that should be closed would correspond to facilities that have an amount of supply to give out. On the other hand, the empty slots from those facilities that should be opened but do not have enough clients to reach B would correspond to clients in CFL, which have to be satisfied by the supply from other facilities. Of course, we do not know in advance which facilities should be opened and which should be closed, but the reduction does not require this knowledge. In order to avoid a confusion of terminology arising from the reversal of the client-facility roles, we say that the instance of CFL has *supply points* (facilities), each with some *total supply* (capacity), and *demand points* (clients), each with some amount of demand. The goal is to *select* (open) some supply points, paying a *selection cost* (facility opening cost), and to assign each demand point to a selected supply point, paying a connection cost, so that each supply point serves at most the amount of demand equal to its total supply.

The CFL instance \mathcal{I}_{cap} that we create is defined as follows (see Figure 3). For each facility $i \in \mathcal{F}_2$ that has $n_i \leq B$ clients, create a supply point at its location with total supply B and selection cost $\delta \cdot n_i \cdot l(i)$, where $l(i)$ is the distance between i and its closest other facility $i' \in \mathcal{F}_2$, $i' \neq i$, and δ is a constant to be optimized later. In addition, create a demand point at this location, with

demand $B - n_i$. This is the additional number of clients that this facility would need in order to reach B . If a facility i has more than B clients, $n_i > B$, then \mathcal{I}_{cap} will have two supply points at this location, and no demand points. The first supply point has cost 0 and total supply $n_i - B$, and the second supply point has total supply B and cost $\delta \cdot B \cdot l(i)$, analogously to the previous case. The distances of \mathcal{I}_{cap} are the same as in \mathcal{I}_2 .

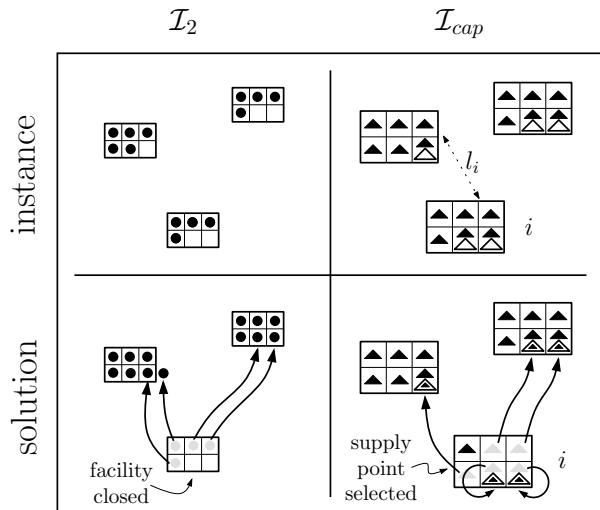


Figure 3: The top row shows the correspondence between the instance \mathcal{I}_2 of LBFL (with $B = 6$) and the constructed instance of CFL, \mathcal{I}_{cap} . The circles represent the clients in the LBFL problem. The black triangles represent the amount of supply at a supply point, and the white triangles represent the amount of demand. The bottom row shows the correspondence between the solutions to these instances. The location i which is closed in the solution to \mathcal{I}_2 is selected in the solution to \mathcal{I}_{cap} . Three units of its supply satisfy the demand of other locations, and two units of supply satisfy the demand of the same location.

We now bound the cost of the optimal solution to \mathcal{I}_{cap} in terms of the optimal solution to \mathcal{I}_2 .

LEMMA 4.1. $OPT(\mathcal{I}_{cap}) \leq (1 + \delta) \cdot OPT(\mathcal{I}_2)$

Proof. Let us examine the form of the optimal LBFL solution to \mathcal{I}_2 , and then use it to construct a specific solution for \mathcal{I}_{cap} , whose cost is then an upper bound on $OPT(\mathcal{I}_{cap})$. We can assume without loss of generality, by using the triangle inequality, that in the solution to \mathcal{I}_2 , there is no facility i such that some clients are assigned from another facility to i , and other clients are assigned from i to another facility.

The solution that we propose for the CFL instance \mathcal{I}_{cap} corresponds to this LBFL solution in the following way (see Figure 3). We select the supply points corresponding to the facilities which are closed in the LBFL solution, and let them satisfy all the demand at their own locations. The supply points with cost 0 at the locations with $n_i > B$ are selected as well. Whenever k clients are assigned from facility i to facility i' by the LBFL solution, we say that k amount of supply is sent from the supply point i to the demand point i' . The resulting solution may send more supply to a location than this location's demand, but this can be easily corrected without increasing the cost.

To see that a feasible solution to the CFL instance \mathcal{I}_{cap} is obtained, we make two observations. First, all the demand of \mathcal{I}_{cap} is satisfied: if a location i with demand $B - n_i$ is opened by the LBFL solution (like the top two facilities in Figure 3), then there must be at least $B - n_i$ additional clients assigned to it in order to satisfy the lower bound requirement, which means that in the corresponding solution to \mathcal{I}_{cap} , all the demand of i is satisfied by supply from other locations; if i is closed (like the bottom facility in Figure 3), then it will be selected, and will be able to satisfy its own demand of $B - n_i$ using part of its total supply of B . The second observation is that the selected supply points do not exceed their total supply when satisfying the demands assigned to them by this solution. A selected supply point i sends out the amount of supply at most equal to the number of clients leaving the corresponding facility in the solution to LBFL. But if this facility is closed, then it is sending n_i clients elsewhere, which is equal to the total supply of i in the case that $n_i > B$, or otherwise is equal to its total supply, B , minus the $B - n_i$ amount that it uses to satisfy its own demand. If the facility is open, then the only case in which it is sending out clients is if it started with $n_i > B$, and is sending out at most $n_i - B$, which is equal to the total supply of its corresponding supply point of cost 0.

Now we bound the cost of the constructed solution to \mathcal{I}_{cap} . Its connection cost is at most $OPT(\mathcal{I}_2)$, since we only moved supply that corresponds to clients that are reassigned in the solution to \mathcal{I}_2 . The solution's selection cost is at most δ times the connection cost of \mathcal{I}_2 , because the selection cost of $\delta \cdot l(i) \cdot \min(n_i, B)$ is paid for each supply point i that corresponds to a closed facility, and the LBFL solution has to pay at least $n_i \cdot l(i)$ in connection cost in order to move the n_i clients from the closed facility i to other facilities, whose distance from i is at least $l(i)$. Thus the total cost of our solution is at most $(1 + \delta) \cdot OPT(\mathcal{I}_2)$. \square

The next step of the algorithm is to solve the CFL instance \mathcal{I}_{cap} , obtaining a solution \mathcal{S}_{cap} , by using one

of the known constant-factor approximation algorithms for it (e.g. [21]). Say that the approximation ratio for this algorithm is γ . Then we get the following corollary to Lemma 4.1.

COROLLARY 4.1. *The cost of the solution \mathcal{S}_{cap} found for the instance \mathcal{I}_{cap} is at most $(1 + \delta)\gamma \cdot OPT(\mathcal{I}_2)$.*

5 Reassignment of clients

Once the CFL instance \mathcal{I}_{cap} is solved, we reassign clients from their locations in \mathcal{I}_2 according to the obtained solution \mathcal{S}_{cap} , in a way that we explain and analyze in this section. We assume without loss of generality that in \mathcal{S}_{cap} , if a location is selected, then it satisfies its own demand.

The first type of reassignment of clients that we perform is exactly as proposed by the solution \mathcal{S}_{cap} : if the demand at some location i' is satisfied by the supply from another location i in \mathcal{S}_{cap} , then we move the number of clients equal to this supply from i to i' . It is always possible to perform this reassignment because the amount of supply exported from i is never more than its number of clients, n_i . This is true because either $n_i \geq B$, in which case the total supply of i is equal to its number of clients, or else $n_i < B$, in which case $B - n_i$ amount of supply is used to satisfy i 's own demand, and only n_i amount remains for export.

The reason that we do not yet have a feasible solution to the LBFL problem is the following. The specification of the CFL problem requires that any feasible solution satisfy all of the clients (demands); however, it does not require that an opened facility (selected supply point) use *all* of its capacity (supply). As a result, we may now have facilities, whose supply points are selected, but not all of whose clients are reassigned elsewhere. For example, if reassignment is performed based on the CFL solution in the bottom-right section of Figure 3, then out of four clients at facility i , three would be moved to other facilities, but one would be left. The rest of this section explains how our algorithm deals with these clients that remain at the selected facilities. Let us summarize the two types of facilities that result after the first reassignment.

- There are some facilities, call this set $\mathcal{A} \subseteq \mathcal{F}_2$, which now have at least B clients. This set includes all facilities whose corresponding supply points are not selected by \mathcal{S}_{cap} (and therefore whose demand amount of $B - n_i$, if positive, is fulfilled by supply from other locations).
- There are other facilities, $\bar{\mathcal{A}} = \mathcal{F}_2 \setminus \mathcal{A}$, which now have less than B clients. The way this happens is that their corresponding supply points are selected

by \mathcal{S}_{cap} , and (possibly) some of their clients are reassigned to other locations. Note that for each such facility $i \in \overline{\mathcal{A}}$, a selection cost of $\delta \cdot l(i) \cdot \min(n_i, B)$ is paid by the solution \mathcal{S}_{cap} .

Facilities in the set \mathcal{A} constitute the easy case, as we just open them and let them serve the clients currently assigned to them, satisfying the lower bound requirement. For the other facilities, however, we have to do a little more work.

Let us construct a directed graph G whose nodes are the facilities of \mathcal{F}_2 . For each facility $i \in \overline{\mathcal{A}}$, include an edge (i, i') , where $i' \in \mathcal{F}_2$ is the nearest neighbor of i (remember that the distance between i and i' is $l(i)$). When constructing this graph, we use some ordering on the facilities to break ties and avoid cycles in the graph. As a result, G will consist of two types of connected components:

1. A tree, whose root is in \mathcal{A} , and whose edges are directed toward the root.
2. A tree containing exactly one double edge (i.e. the pair of closest nodes with edges in both directions between them), with other edges of the tree directed toward this double edge.

Note that the facilities from \mathcal{A} are always roots of type-1 trees, or singletons (which is a special case), as they do not have out-edges. Facilities from $\overline{\mathcal{A}}$ make up the non-root nodes of type-1 trees and the type-2 trees entirely. In particular, they are always in components of size at least two, which is important for our algorithm.

We now use the graph G to make some more reassignments of clients, to make sure that the lower bound constraints are satisfied. For each component of type 1, we do the following procedure on each facility i in this component, bottom-up (see Figure 4). If i has at least B clients, then open facility i and cut the tree edge going up from i . If i has less than B clients, then send all of these clients from i to its parent facility in the tree. Since the root is in \mathcal{A} , it will always have at least B clients, and already be open. Thus at the end of this procedure, each facility in the processed component will have either 0 or at least B clients, satisfying the lower bound constraints. Also notice that during this process, we send strictly less than B clients on each edge of the component.

For the second type of component, we perform the same bottom-up procedure on the parts of the tree directed toward the double edge. The only difference is in what to do with the double edge itself, whose endpoints we call i_1 and i_2 . Here we consider several cases. If each of i_1 and i_2 has at least B clients, then open both of them. If one of them, say i_1 , has at

least B clients, and i_2 has less than B , then transfer all clients from i_2 to i_1 and open i_1 . If each of them has less than B , but in total the two of them have at least B , then we transfer all clients from i_2 to i_1 and open i_1 . In the case that the total number of clients at i_1 and i_2 is less than B , we find the closest facility $i \in \mathcal{A}$ to either one of the two endpoints (i.e., one minimizing $\min(c(i, i_1), c(i, i_2))$). Let us say without loss of generality that i is closer to i_1 . Then we send clients from i_2 to i_1 , and then all of them from i_1 to i . Since $i \in \mathcal{A}$, it already has at least B clients and is open, so the procedure overall produces a feasible solution to LBFL, which is the final solution that we output.

What remains to be done is to bound the cost incurred by all the transfers of clients that are performed after the solution of \mathcal{I}_{cap} . We bound it in terms of the connection cost, C^{cap} , and the selection cost, F^{cap} , of our CFL solution \mathcal{S}_{cap} .

LEMMA 5.1. *The cost of the solution found by our algorithm for \mathcal{I}_2 is at most*

$$\frac{2\alpha}{2\alpha - 1} \cdot C^{cap} + \frac{1}{\delta\alpha} \cdot F^{cap} \leq \max\left(\frac{2\alpha}{2\alpha - 1}, \frac{1}{\delta\alpha}\right) \cdot \text{cost}(\mathcal{S}_{cap}).$$

Proof. After solving \mathcal{I}_{cap} , the algorithm makes three types of client reassignments, for which we bound the costs separately:

1. Reassign clients according to the supply and demand assignments of the solution \mathcal{S}_{cap} .
2. Reassign at most B clients for each edge of the graph G .
3. In case that facilities i_1 and i_2 forming a double edge in G don't have a total of B clients, reassign at most B clients from i_1 to the nearest facility $i \in \mathcal{A}$.

Reassignment of type 1 costs at most C^{cap} , as connection costs of \mathcal{I}_{cap} are the same as those of \mathcal{I}_2 .

For the second type of reassignment, we notice that for each edge in G which starts at a facility i and has length $l(i)$, the solution \mathcal{S}_{cap} has paid $\delta \cdot l(i) \cdot \min(n_i, B)$ as a selection cost for the supply point i . But since \mathcal{I}_2 came from a bicriteria solution with parameter α , we know that $n_i \geq \alpha B$. So for each edge in G , the selection cost F^{cap} includes an amount of at least $\delta \cdot l(i) \cdot \alpha B$, whereas we pay at most $l(i) \cdot B$ for transferring clients on this edge. Thus, the total cost of reassignments of type 2 is at most $F^{cap}/\delta\alpha$.

For the third type of reassignment, we bound its cost against the connection cost of \mathcal{S}_{cap} . In particular, we make the following observation about the facilities i_1 and i_2 forming the double edge in G . As a result of the

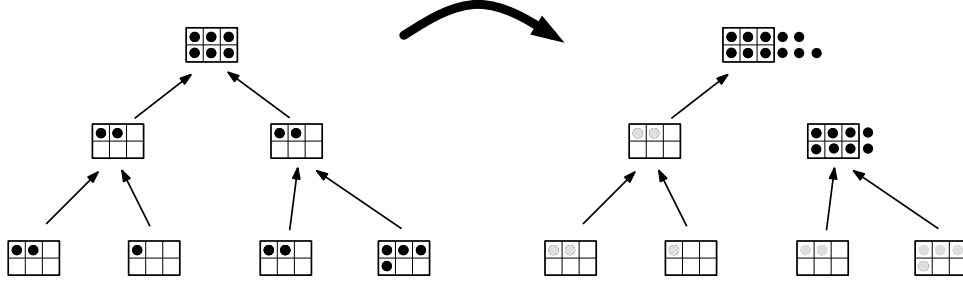


Figure 4: The outcome of the bottom-up procedure of client reassignment, with $B = 6$, on a connected component of type 1.

bicriteria algorithm, each of them has at least αB clients in \mathcal{I}_2 , and so together they have at least $2\alpha B > B$ clients (since $\alpha > \frac{1}{2}$). However, after the solution of the CFL and reassignments of type 1 and 2, they have less than B . Since the bottom-up reassignment on the edges of G could have only added clients to i_1 and i_2 , it must be that at least $(2\alpha - 1)B$ clients were moved to facilities in \mathcal{A} (which are all at least as far as i) by the first kind of reassignment. Therefore, for each such pair i_1 and i_2 that sends clients to its nearest facility $i \in \mathcal{A}$, the solution \mathcal{S}_{cap} to our CFL instance must have paid at least $(2\alpha - 1)B \cdot c(i_1, i)$ in connection cost. So the total cost of type-3 reassignments is at most $C^{cap}/(2\alpha - 1)$. Adding the bounds, we get the result. \square

By combining Lemma 5.1, Corollary 4.1, and Lemma 3.3, we get the following final result.

THEOREM 5.1. *There is a constant-factor approximation algorithm for the lower-bounded facility location problem.*

Proof. Setting $\delta = \frac{2\alpha-1}{2\alpha^2}$ and using it in Lemma 5.1 shows that our solution costs at most $\frac{2\alpha}{2\alpha-1}$ times the solution to the CFL instance \mathcal{I}_{cap} . Then applying Corollary 4.1 we get that it is a $\beta = \frac{2\alpha}{2\alpha-1}(1 + \frac{2\alpha-1}{2\alpha^2})\gamma$ factor approximation for the instance \mathcal{I}_2 , which can then be used in Lemma 3.3. Using the value of $\alpha = 0.68$, the $\rho = 1.52$ approximation algorithm for FL [16], and $\gamma = (5.83 + \varepsilon)$ approximation algorithm for CFL [21], the overall approximation ratio becomes $558 + \varepsilon$. \square

6 Extensions and conclusions

In this paper we have presented the first constant-factor true approximation algorithm for the lower-bounded facility location problem. The constant in the approximation guarantee that we obtain is of course not practical, so the main contribution of our work is a theoretical demonstration that there exist polynomial-time constant-factor approximation algorithms which solve

the LBFL problem without violating the constraints. It would be interesting to find algorithms with much better guarantees, which may be useful in practice, and we leave it for future work. The running time of our algorithm is dominated by the single call to the capacitated facility location subroutine, since both the initial bicriteria solution and the final client reassignments can be found efficiently. The known algorithms for CFL, on the other hand, use local search, which tends to have rather high theoretical bounds for the worst-case running time.

Our algorithm can be extended to work for the case of clients with non-unit demands, in which each client has a non-negative demand $d(j)$, the connection cost for client j becomes scaled by $d(j)$, and the lower-bound constraints now require that the total demand served by a facility is at least B . However, the known solutions for capacitated facility location all allow the splitting of a client's demand, with parts of it being assigned to different facilities. So because we make use of the algorithms for CFL, our algorithm would also have to allow this kind of splitting of demand. We note that for the case of CFL, it is reasonable to allow splittable demand because otherwise it is NP-hard to even determine the existence of a feasible solution. Although this is not true for LBFL, and the feasibility question is easy even with unsplittable demands, one can show that this problem is not approximable to any factor that is independent of the demand values, unless $P = NP$.

Unfortunately, the algorithm presented in this paper does not extend to the generalization of the LBFL problem in which each facility i has its own lower bound B_i for the number of clients that it has to serve if opened. The only step that fails to extend to such non-uniform bounds is the proof of Lemma 3.2, transforming the original instance to \mathcal{I}_2 : the clients now cannot just be moved to the closest facility $i' \in \mathcal{F}_2$, because that may violate its bound $B_{i'}$. We leave the solution of LBFL with non-uniform bounds to future work.

In fact, we have a simple reduction that shows how to use the solution to the non-uniform LBFL in order to solve a variant of the universal facility location problem with monotone *non-increasing* facility costs (as opposed to the monotone non-decreasing costs which have been considered so far), without any loss in the approximation guarantee. This version of universal facility location generalizes LBFL. The reduction just involves creating multiple facilities in place of each original facility, with appropriate costs and lower bounds, but requires that the LBFL problem be solved with a true approximation, and not in the bicriteria sense. Another interesting related open problem is the universal facility location with *non-monotone* costs.

7 Acknowledgements

I thank Elliot Anshelevich for suggesting this problem to me, Éva Tardos and David Shmoys for helpful discussions, and Yogi Sharma for discussions of the problem and a critical reading of the manuscript. I also thank the anonymous referees for their comments and suggestions.

References

- [1] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 21–29, 2001.
- [2] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proc. 40th IEEE Symp. on Foundations of Computer Science*, pages 378–388, 1999.
- [3] F. Chudak and D. Shmoys. Improved approximation algorithms for a capacitated facility location problem. In *Proc. 10th ACM Symp. on Discrete Algorithms*, pages 875–876, 1999.
- [4] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2003.
- [5] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. In *IPCO*, pages 99–113, 1999.
- [6] N. Garg, R. Khandekar, and V. Pandit. Improved approximation for universal facility location. In *Proc. 16th ACM Symp. on Discrete Algorithms*, pages 959–960, 2005.
- [7] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. In *Proc. 41st IEEE Symp. on Foundations of Computer Science*, page 603, 2000.
- [8] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problems. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 383–388, 2001.
- [9] M. T. Hajiaghayi, M. Mahdian, and V. S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42:42–47, 2003.
- [10] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 731–740, 2002.
- [11] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- [12] D. R. Karger and M. Minkoff. Building steiner trees with incomplete global knowledge. In *Proc. 41st IEEE Symp. on Foundations of Computer Science*, page 613, 2000.
- [13] 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.
- [14] A. Lim, F. Wang, and Z. Xu. A transportation problem with minimum quantity commitment. *Transportation Science*, 40(1):117–129, 2006.
- [15] M. Mahdian and M. Pál. Universal facility location. In *European Symposium on Algorithms*, pages 409–421, 2003.
- [16] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Proc. 5th APPROX*, pages 229–242, 2002.
- [17] M. Mahdian, Y. Ye, and J. Zhang. A 2-approximation algorithm for the soft-capacitated facility location problem. In *Proc. 6th APPROX*, pages 129–140, 2003.
- [18] M. Pal, E. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, page 329, 2001.
- [19] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *Proc. 29th ACM Symp. on Theory of Computing*, pages 265–274, 1997.
- [20] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *IPCO*, pages 240–257, 2002.
- [21] J. Zhang, B. Chen, and Y. Ye. A multiexchange local search algorithm for the capacitated facility location problem. *Math. Oper. Res.*, 30(2):389–403, 2005.