SIAM J. COMPUT.
Vol. 39, No. 6, pp. 2212–2231

AN OPTIMAL BIFACTOR APPROXIMATION ALGORITHM FOR THE METRIC UNCAPACITATED FACILITY LOCATION PROBLEM*

JAROSLAW BYRKA† AND KAREN AARDAL‡

Abstract. We obtain a 1.5-approximation algorithm for the metric uncapacitated facility location (UFL) problem, which improves on the previously best known 1.52-approximation algorithm by Mahdian, Ye, and Zhang. Note that the approximability lower bound by Guha and Khuller is $1.463\ldots$. An algorithm is a (λ_f,λ_c)-*approximation algorithm* if the solution it produces has total cost at most $\lambda_f \cdot F^* + \lambda_c \cdot C^*$, where F^* and C^* are the facility and the connection cost of an optimal solution. Our new algorithm, which is a modification of the $(1 + 2/e)$-approximation algorithm of Chudak and Shmoys, is a $(1.6774, 1.3738)$-approximation algorithm for the UFL problem and is the first one that touches the approximability limit curve $(\gamma_f, 1 + 2e^{-\gamma_f})$ established by Jain, Mahdian, and Saberi. As a consequence, we obtain the first optimal approximation algorithm for instances dominated by connection costs. When combined with a $(1.11, 1.7764)$-approximation algorithm proposed by Jain et al., and later analyzed by Mahdian et al., we obtain the overall approximation guarantee of 1.5 for the metric UFL problem. We also describe how to use our algorithm to improve the approximation ratio for the 3-level version of UFL.

Key words. facility location, approximation algorithms, LP-rounding

AMS subject classifications. 90B80, 68W25, 68W40, 68W20

DOI. 10.1137/070708901

1. Introduction.

1.1. Background on uncapacitated facility location. The uncapacitated facility location (UFL) problem is defined as follows. We are given a set $\mathcal{F}$ of *facilities* and a set $\mathcal{C}$ of *clients*. For every facility $i \in \mathcal{F}$, there is a nonnegative number f_i denoting the *opening cost* of the facility. Furthermore, for every client $j \in \mathcal{C}$ and facility $i \in \mathcal{F}$, there is a *connection cost* c_{ij} between facility i and client j. The goal is to open a subset of the facilities $\mathcal{F}' \subseteq \mathcal{F}$ and connect each client to an open facility so that the total cost is minimized. The UFL problem is NP-complete and max SNP-hard (see [14]). A UFL instance is *metric* if its *connection cost* function satisfies the following variant of the *triangle inequality*:

$$c_{ij} \le c_{ij'} + c_{i'j'} + c_{i'j} \text{ for any } i, i' \in \mathcal{C} \text{ and } j, j' \in \mathcal{F}. \tag{1.1}$$

We will say that an algorithm is a λ-approximation algorithm for a minimization problem if it computes, in polynomial time, a solution that is at most λ times more expensive than the optimal solution. Specifically, for the UFL problem we consider the notion of *bifactor approximation* introduced by Charikar and Guha [7, 8]. We say that an algorithm is a (λ_f,λ_c)-approximation algorithm if the solution it delivers has total

*Received by the editors November 21, 2007; accepted for publication (in revised form) December 28, 2009; published electronically March 17, 2010. A short version of this paper appeared in the Proceedings of APPROX'07 [5]. The results reported in this paper were obtained while both authors were affiliated with CWI, Amsterdam.

http://www.siam.org/journals/sicomp/39-6/70890.html

†Institute of Mathematics, EPFL, CH-1015 Lausanne, Switzerland (Jaroslaw.Byrka@epfl.ch). The work of this author was partially supported by the EU Marie Curie Research Training Network ADONET, contract MRTN-CT-2003-504438.

‡Delft Institute of Applied Mathematics, Delft University of Technology, 2628CD Delft, The Netherlands (k.i.aardal@tudelft.nl), and Centrum Wiskunde & Informatica, Amsterdam, The Netherlands. The work of this author was partially supported by the Dutch BSIK/BRICKS project.

cost at most $\lambda_f \cdot F^* + \lambda_c \cdot C^*$, where F^* and C^* denote, respectively, the facility and the connection cost of an optimal solution. Note the potential ambiguity resulting from the possible existence of multiple optimal solutions. When presenting our algorithm, we will compare the solution cost only to the cost of the initial fractional solution. Nevertheless, as we observe at the end of section 4, adding an additional scaling step to our algorithm is sufficient to get a worst-case guarantee in a comparison with any feasible fractional solution.

Guha and Khuller [14] proved by a reduction from set cover that there is no polynomial time λ-approximation algorithm for the metric UFL problem with $\lambda < 1.463$, unless $NP \subseteq DTIME(n^{\log\log n})$. Sviridenko showed that the approximation lower bound of 1.463 holds, unless $P = NP$ (see [25]). Jain, Mahdian, and Saberi [18] generalized the argument of Guha and Khuller to show that the existence of a (λ_f,λ_c)-approximation algorithm with $\lambda_c < 1 + 2e^{-\lambda_f}$ would imply $NP \subseteq DTIME(n^{\log\log n})$.

The UFL problem has a rich history starting in the 1960s. The first results on approximation algorithms are due to Cornuéjols, Fisher, and Nemhauser [11], who considered the problem with an objective function of maximizing the "profit" of connecting clients to facilities, minus the cost of opening facilities. They showed that a greedy algorithm gives an approximation ratio of $(1 - 1/e) = 0.632\ldots$, where e is the base of the natural logarithm. This ratio was later improved to 0.828 by Ageev and Sviridenko [2].

For the objective function of minimizing the sum of connection cost and opening cost, Hochbaum [17] presented a greedy algorithm with an $O(\log n)$-approximation guarantee, where n is the number of clients. By a straightforward reduction from the set cover problem, it can be shown that this cannot be improved unless $NP \subseteq DTIME[n^{O(\log\log n)}]$ due to a result by Feige [12]. However, if the connection costs are restricted to satisfying the triangle inequality (1.1), then constant approximation guarantees can be obtained. In all results mentioned below, except for the maximization objectives, it is assumed that the costs satisfy these restrictions. If the distances between facilities and clients are Euclidean, then for some location problems approximation schemes have been obtained [4].

The first approximation algorithm with constant approximation ratio for the metric minimization problem was developed by Shmoys, Tardos, and Aardal [23]. Since then numerous improvements have been made. Guha and Khuller [14, 15] introduced a *greedy augmentation procedure* (see also Charikar and Guha [7, 8]). A series of approximation algorithms based on linear programming (LP) rounding was then developed (see, e.g., [9, 10, 24]). There are also greedy algorithms that only use the LP-relaxation implicitly to obtain a lower bound for a primal-dual analysis. An example is the JMS 1.61-approximation algorithm developed by Jain, Mahdian, and Saberi [18]. Some algorithms combine several techniques, like the 1.52-approximation algorithm of Mahdian, Ye, and Zhang [20, 21], which uses the JMS algorithm and the greedy augmentation procedure. Up to now, their approximation ratio of 1.52 was the best known. Many more algorithms have been considered for the UFL problem and its variants. We refer the interested reader to survey papers by Shmoys [22] and Vygen [25].

1.2. Some basic techniques. In several LP-based approximation algorithms a *clustering step* is part of an algorithm for creating a feasible solution; see section 2.2 for more details. In this step a not-yet-clustered client is chosen as the so-called cluster center, and one of the facilities that fractionally serves the cluster center, in the LP-solution is opened. Our main technique is to modify the support graph corre-

sponding to the LP-solution before clustering, and to use various average distances in the fractional solution to bound the cost of the obtained solution.

A similar way of modifying the LP-solution, called *filtering*, was introduced by Lin and Vitter [19]. Lin and Vitter considered a broad class of 0-1 problems having both covering and packing constraints. They start by solving the LP-relaxation of the problem, and in the subsequent filtering step they select a subset of the variables that have positive value in the LP-solution and that have relatively large objective coefficients. These variables are set equal to zero, which results in a modified problem. The LP-relaxation of this modified problem is then solved and rounding is applied. In the paper by Shmoys, Tardos, and Aardal [23] filtering was also used in order to bound the connection costs. Here again a subset of the variables that have a positive value in the LP-solution are set equal to zero. The remaining positive variables were scaled so as to remain feasible for the original LP-relaxation.

Later, Chudak [9] observed that the LP-relaxation was already filtered in a certain sense as it is possible to state that if a client is fractionally connected to a facility in the LP-solution, then one can bound the cost of this connection in terms of the optimal LP-dual variables. This observation was later used by Aardal, Chudak, and Shmoys [1] in their algorithm for multilevel problems, and by Sviridenko [24]. The filtering done in our algorithm is slightly different, as the filtered LP-solution is not necessarily feasible with respect to the LP-relaxation. Throughout this paper we will use the term *sparsening technique* for the combination of filtering with our new analysis.

1.3. Our contribution. We modify the $(1 + 2/e)$-approximation algorithm of Chudak [9] (see also Chudak and Shmoys [10]) to obtain a new $(1.6774, 1.3738)$-approximation algorithm for the UFL problem. Our LP-rounding algorithm is the first one that achieves an optimal bifactor approximation due to the matching lower bound of $(\lambda_f, 1 + 2e^{-\lambda_f})$ established by Jain, Mahdian, and Saberi [18]. In fact we obtain an algorithm for each point $(\lambda_f, 1 + 2e^{-\lambda_f})$ such that $\lambda_f \geq 1.6774$, which means that we have an optimal approximation algorithm for instances dominated by connection cost (see Figure 1.1).

One of the main technical contributions of the paper is the proof of Lemma 3.3, which gives a bound on the expected connection cost in the case of using a path via a cluster center to connect a client. This lemma may potentially be useful in constructing new algorithms for UFL and related problems.

One could view our contribution as an improved analysis of a minor modification of the algorithm by Sviridenko [24], which also introduces filtering to the algorithm of Chudak and Shmoys. The filtering process that is used both in our algorithm and in the algorithm by Sviridenko is relatively easy to describe, but the analysis of the impact of this technique on the quality of the obtained solution is quite involved in each case. Therefore, we prefer to state our algorithm as an application of the sparsening technique to the algorithm of Chudak and Shmoys, which in our opinion is relatively easy to describe and analyze.

We start by observing that for a certain class of instances the analysis of the algorithm of Chudak and Shmoys may be improved. We call these instances *regular*, and for the other instances we propose a measure of their *irregularity*. The goal of the sparsening technique is to explore the irregularity of instances that are potentially tight for the original algorithm of Chudak and Shmoys. We cluster the given instance in the same way as in the 1.58-approximation algorithm by Sviridenko [24], but we continue our algorithm in the spirit of Chudak and Shmoys' algorithm, and we use

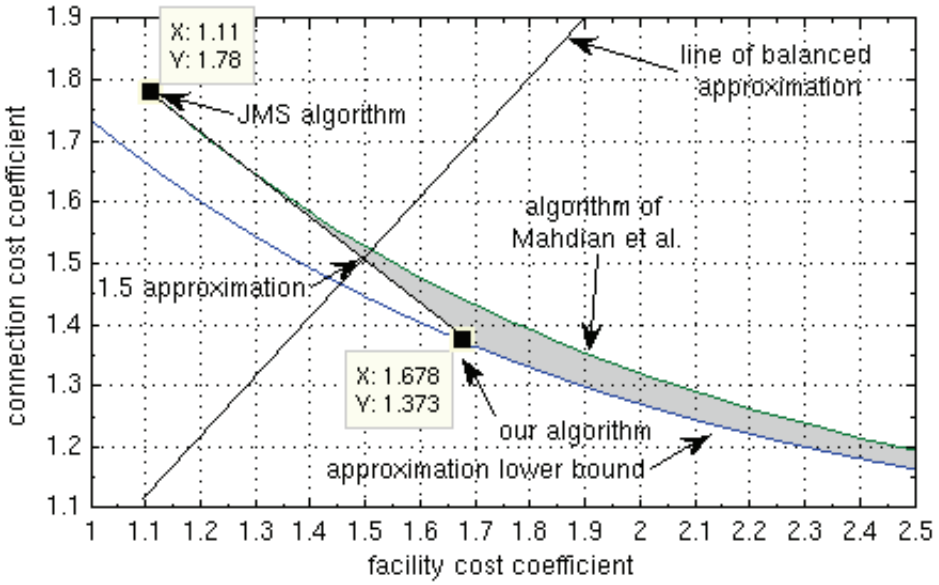

FIG. 1.1. *Bifactor approximation picture. The gray area corresponds to the improvement due to our algorithm.*

certain average distances to control the irregularities, which leads to an improved bifactor approximation guarantee.

Our new algorithm may be combined with the $(1.11, 1.7764)$-approximation algorithm of Jain et al. to obtain a 1.5-approximation algorithm for the UFL problem. This is an improvement over the previously best known 1.52-approximation algorithm of Mahdian et al., and it cuts off 1/3 of the gap with the approximation lower bound by Guha and Khuller [14]. An earlier version of this paper appeared in [5].

We now give an informal sketch of our algorithm. Using this description, we give an outline of the paper.

Sketch of the algorithm.

1. Solve the LP-relaxation of the problem.
2. Modify the fractional solution by
   - scaling up the facility opening variables;
   - modifying the connection variables to completely use the "closest" fractionally open facilities;
   - splitting facilities, if necessary, such that there is no slack between the amount that a client is assigned to a facility and the amount by which this facility is opened.
3. Divide clients into clusters based on the current fractional solution. In each cluster, a specific client is assigned to be a "cluster center."
4. For every cluster, open one of the "close" facilities of the cluster center.
5. For each facility not considered above, open it independently with probability equal to the fractional opening.
6. Connect each client to an open facility that is closest to it.

In section 2 we give a brief overview of the main ingredients of some known approximation algorithms for the UFL problem. In particular we state the LP-relaxation of

UFL and describe clustering, scaling, and greedy augmentation. The clustering technique is common for the existing LP-rounding algorithms for the UFL problem, and it is applied in steps 3 and 4 of the above algorithm. Sparsening of the support graph of the LP-solution, which is the essence of step 2, is discussed in section 3, where we also prove the crucial lemma on certain connection costs. A more detailed description of the algorithm and its analysis are presented in section 4, and the 1.5-approximation algorithm is stated in section 5. In section 6 we show that the new $(1.6774, 1.3738)$-approximation algorithm may also be used to improve the approximation ratio for the 3-level version of the UFL problem to 2.492. A randomized approach to clustering is discussed in section 7, and, finally, in section 8 we present some concluding remarks and open problems.

2. Preliminaries. We will review the concept of LP-rounding algorithms for the metric UFL problem. These are algorithms that first solve the linear relaxation of a given integer programming (IP) formulation of the problem, and then round the fractional solution to produce an integer solution with a value not too much higher than the starting fractional solution. Since the optimal fractional solution is at most as expensive as an optimal integral solution, we obtain an estimation of the approximation factor.

2.1. IP formulation and relaxation. The UFL problem has a natural formulation as the following IP-problem:

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{F},\, j\in\mathcal{C}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i \\
\text{s.t.} \quad & \sum_{i\in\mathcal{F}} x_{ij} = 1 && \text{for all } j \in \mathcal{C}, \\
& x_{ij} - y_i \le 0 && \text{for all } i \in \mathcal{F},\ j \in \mathcal{C}, \\
& x_{ij}, y_i \in \{0,1\} && \text{for all } i \in \mathcal{F},\ j \in \mathcal{C}.
\end{aligned} \tag{2.1}$$

A linear relaxation of this IP formulation is obtained by replacing the integrality constraints (2.1) by the constraint $x_{ij} \ge 0$ for all $i \in \mathcal{F}$, $j \in \mathcal{C}$. The value of the solution to this LP-relaxation will serve as a lower bound for the cost of the optimal solution. We will also make use of the following dual formulation of this LP:

$$\begin{aligned}
\max \quad & \sum_{j\in\mathcal{C}} v_j \\
\text{s.t.} \quad & \sum_{j\in\mathcal{C}} w_{ij} \le f_i && \text{for all } i \in \mathcal{F}, \\
& v_j - w_{ij} \le c_{ij} && \text{for all } i \in \mathcal{F},\ j \in \mathcal{C}, \\
& w_{ij} \ge 0 && \text{for all } i \in \mathcal{F},\ j \in \mathcal{C}.
\end{aligned}$$

2.2. Clustering. The first constant factor approximation algorithm for the metric UFL problem by Shmoys et al., but also the algorithms by Chudak and Shmoys and by Sviridenko, are based on the following clustering procedure. Suppose we are given an optimal solution to the LP-relaxation of our problem. Consider the bipartite graph $G = ((V', V''), E)$ with vertices V' being the facilities and V'' the clients of the instance, and where there is an edge between a facility $i \in V'$ and a client $j \in V''$ if the corresponding variable x_{ij} in the optimal solution to the LP-relaxation is positive. We call G a *support graph* of the LP-solution. If two clients are both adjacent to the same facility in graph G, we will say that they are *neighbors* in G.

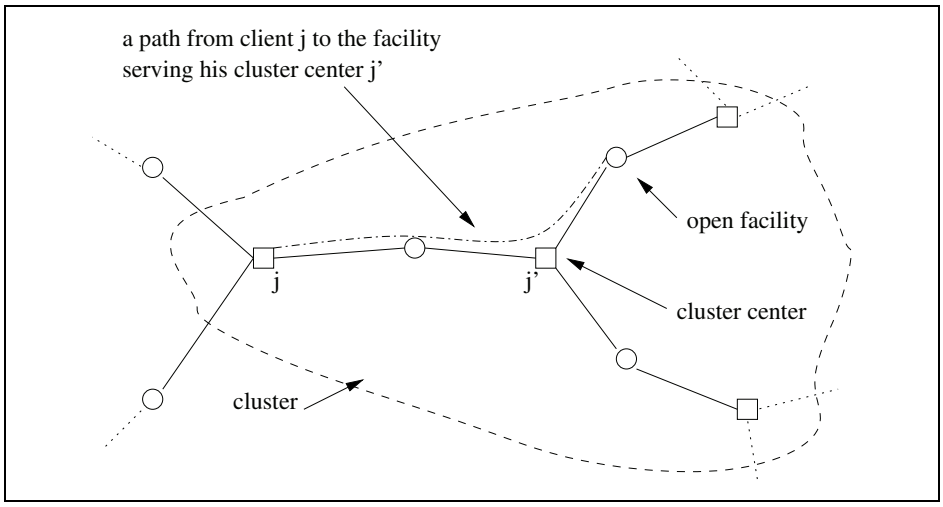

FIG. 2.1. *A cluster. If we make sure that at least one facility is open close to a cluster center j', then any other client j from the cluster may use this facility. Because the connection costs are assumed to be metric, the distance to this facility is at most the length of the shortest path from j to the open facility.*

The clustering in this graph is a partitioning of clients into clusters together with a choice of a leading client for each of the clusters. This leading client is called a *cluster center*. Additionally we require that no two cluster centers be neighbors in the support graph. This property helps us to open one of the adjacent facilities for each cluster center. For a picture of a cluster see Figure 2.1.

The algorithms by Shmoys et al., Chudak and Shmoys, and Sviridenko all use the following procedure to obtain the clustering: While not all the clients are clustered, choose greedily a new cluster center j, and build a cluster from j and all the neighbors of j that are not yet clustered. Obviously the outcome of this procedure is a proper clustering. Moreover, it has a desired property that clients are "close" to their cluster centers. Each of the mentioned LP-rounding algorithms uses a different greedy criterion for choosing new cluster centers. In our algorithm we will use the clustering with the greedy criterion of Sviridenko [24]. Another way of clustering is presented in section 7.

2.3. Scaling and greedy augmentation. The techniques described here are not directly used by our algorithm, but they help to explain why the algorithm of Chudak and Shmoys is close to optimal. We will discuss how scaling facility opening costs before running an algorithm, together with another technique called *greedy augmentation*, may help to balance the analysis of an approximation algorithm for the UFL problem.

The greedy augmentation technique introduced by Guha and Khuller [14] (see also [7, 8]) is as follows. Consider an instance of the metric UFL problem and a feasible solution. For each facility $i \in \mathcal{F}$ that is not opened in this solution, we may compute the amount of cost that is saved by opening facility i, also called the *gain* of opening i, denoted by g_i. While there exists a facility i with positive gain g_i, the greedy augmentation procedure opens a facility that maximizes the ratio of gain to the facility opening cost $\frac{g_i}{f_i}$, and updates the remaining values of g_i.

Suppose we are given an approximation algorithm A for the metric UFL problem and a real number $\delta \geq 1$. Consider the following algorithm $S_\delta(A)$.

ALGORITHM $S_\delta(A)$.

1. Scale up all facility opening costs by a factor of δ;
2. run algorithm A on the modified instance;
3. scale back the opening costs;
4. run the greedy augmentation procedure.

Following the analysis of Mahdian, Ye, and Zhang [20], one may prove the following lemma.

LEMMA 2.1. *Suppose A is a (λ_f,λ_c)-approximation algorithm for the metric UFL problem; then $S_\delta(A)$ is a $(\lambda_f+\ln(\delta), 1+\frac{\lambda_c-1}{\delta})$-approximation algorithm for this problem.*

This method may be applied to balance a (λ_f,λ_c)-approximation algorithm with $\lambda_f \ll \lambda_c$. However, our 1.5-approximation algorithm is balanced differently. It is a composition of two algorithms that have opposite imbalances.

3. Sparsening the graph of the fractional solution. In this section we describe a technique that we use to control the expected connection cost of the obtained integer solution. Our technique is based on the concept of *filtering*, introduced by Lin and Vitter [19]; see section 1.2. We will give an alternative analysis of the effect of filtering on a fractional solution to the LP-relaxation of the UFL problem.

Suppose that, for a given UFL instance, we have solved its LP-relaxation, and that the optimal primal solution is (x^*, y^*) and the corresponding optimal dual solution is (v^*, w^*). Such a fractional solution has facility cost $F^* = \sum_{i\in\mathcal{F}} f_i y_i^*$ and connection cost $C^* = \sum_{i\in\mathcal{F},\, j\in\mathcal{C}} c_{ij} x_{ij}^*$. Each client j has its share v_j^* of the total cost. This cost may again be divided into a client's fractional connection cost $C_j^* = \sum_{i\in\mathcal{F}} c_{ij} x_{ij}^*$ and its fractional facility cost $F_j^* = v_j^* - C_j^*$.

3.1. Motivation and intuition. The idea behind the sparsening technique is to make use of irregularities of an instance if they occur. We call an instance *locally regular* around client j if the facilities that serve j in the fractional solution (x^*, y^*) are all at the same distance from j. An instance that is locally regular around every client is called *regular*. We begin by observing that for such an instance the algorithm of Chudak and Shmoys produces a solution whose cost is bounded by $F^* + (1+\frac{2}{e})C^*$, which is an easy consequence of the original analysis [10], but also follows from our analysis in section 4. Although this observation might not be very powerful itself, the value $F^* + (1+\frac{2}{e})C^*$ happens to be the intersection point between the bifactor approximation lower bound curve $(\lambda_f, 1+2e^{-\lambda_f})$ and the y-axis in Figure 1.1. Moreover, for regular instances we may apply the technique described in section 2.3 to obtain an approximation algorithm corresponding to any single point on this curve. In particular, we may simply use this construction to get an optimal 1.463...-approximation algorithm for regular instances of the metric UFL problem. Note that the proof of the matching hardness of approximation also uses instances that are essentially[1] regular.

The instances that are not regular are called *irregular*, and these are the instances for which it is more difficult to create a feasible integer solution with good bounds on the connection cost. In fractional solutions of irregular instances there exist clients that are fractionally served by facilities at different distances. Our approach is to divide facilities serving a client into two groups, namely, *close* and *distant* facilities.

[1] These instances come from a reduction from the set cover problem. Clients represent elements to be covered, and facilities represent subsets. The distance c_{ij} equals 1 if subset i contains element j, and it equals 3 otherwise. To formally argue about the regularity of such an instance we would need to construct an optimal fractional solution using only facilities at distance 1.

We will remove links to distant facilities before the clustering step, so that if there are irregularities, then distances to cluster centers will decrease.

We measure the local irregularity of an instance by comparing the fractional connection cost of a client to the average distance to its distant facilities. In the case of a regular instance, the sparsening technique gives the same results as the technique described in section 2.3, but for irregular instances sparsening makes it possible to construct an integer solution with a better bound on the connection costs.

3.2. Details. We will start by modifying the optimal fractional LP-solution (x^*, y^*) by scaling the y-variables by a constant $\gamma > 1$ to obtain a fractional solution $(x^*, \tilde{y})$, where $\tilde{y} = \gamma \cdot y^*$. Note that by scaling we might set some $\tilde{y}_i > 1$. In the filtering of Shmoys et al. such a variable would instantly be rounded to 1. However, for the compactness of a later part of our analysis it is important not to round these variables, but rather to split facilities. Before we discuss splitting, let us first modify the connection variables. A version of this argument, which describes all these modifications of the fractional solution at once, is given in [24, Lemma 1].

Suppose that the values of the y-variables are scaled and fixed, but that we now have the freedom to change the values of the x-variables in order to minimize the connection cost. For each client j we compute the values of the corresponding $\tilde{x}$-variables in the following way. We choose an ordering of facilities with *nondecreasing* distances to client j. We connect client j to the first facilities in the ordering so that among the facilities fractionally serving j, only the last one in the chosen ordering may be opened by more than that it serves j. Formally, for any facilities i and i' such that i' is later in the ordering, if $\tilde{x}_{ij} < \tilde{y}_i$, then $\tilde{x}_{i'j} = 0$.

In the next step, we eliminate the occurrences of situations where $0 < \tilde{x}_{ij} < \tilde{y}_i$. We do so by creating an equivalent instance of the UFL problem, where facility i is split into two identical facilities i' and i''. In the new setting, the opening of facility i' is equal to $\tilde{x}_{ij}$ and the opening of facility i'' is equal to $\tilde{y}_i - \tilde{x}_{ij}$. The values of the $\tilde{x}$-variables are updated accordingly. By repeatedly applying this procedure we obtain a so-called *complete* solution $(\overline{x}, \overline{y})$, i.e., a solution in which no pair $i \in \mathcal{F}$, $j \in \mathcal{C}$ exists such that $0 < \overline{x}_{ij} < \overline{y}_i$ (see [24, Lemma 1] for a more detailed argument).

In the new complete solution $(\overline{x}, \overline{y})$ we distinguish groups of facilities that are especially important for a particular client. For a client j we say that a facility i is one of its *close facilities* if it fractionally serves client j in $(\overline{x}, \overline{y})$; $\mathcal{C}_j = \{i \in \mathcal{F} \mid \overline{x}_{ij} > 0\}$ is the set of close facilities of j. If $\overline{x}_{ij} = 0$, but facility i was serving client j in solution (x^*, y^*), then we say that i is a *distant* facility of client j; $\mathcal{D}_j = \{i \in \mathcal{F} \mid \overline{x}_{ij} = 0,\ x^*_{ij} > 0\}$ is the set of distant facilities of j.

We will extensively use the average distances between single clients and groups of facilities defined as follows.

DEFINITION 3.1. *For any client $j \in \mathcal{C}$, and for any subset of facilities $\mathcal{F}' \subset \mathcal{F}$ such that $\sum_{i \in \mathcal{F}'} y^*_i > 0$, let*

$$d(j, \mathcal{F}') = \frac{\sum_{i \in \mathcal{F}'} c_{ij} \cdot y^*_i}{\sum_{i \in \mathcal{F}'} y^*_i}.$$

To interpret differences between certain average distances we will use the following parameter.

DEFINITION 3.2. *Let*

$$r_\gamma(j) = \begin{cases} \frac{d(j,\mathcal{D}_j) - d(j,\mathcal{D}_j \cup \mathcal{C}_j)}{F^*_j} & \text{for } F^*_j > 0, \\ 0 & \text{for } F^*_j = 0. \end{cases}$$

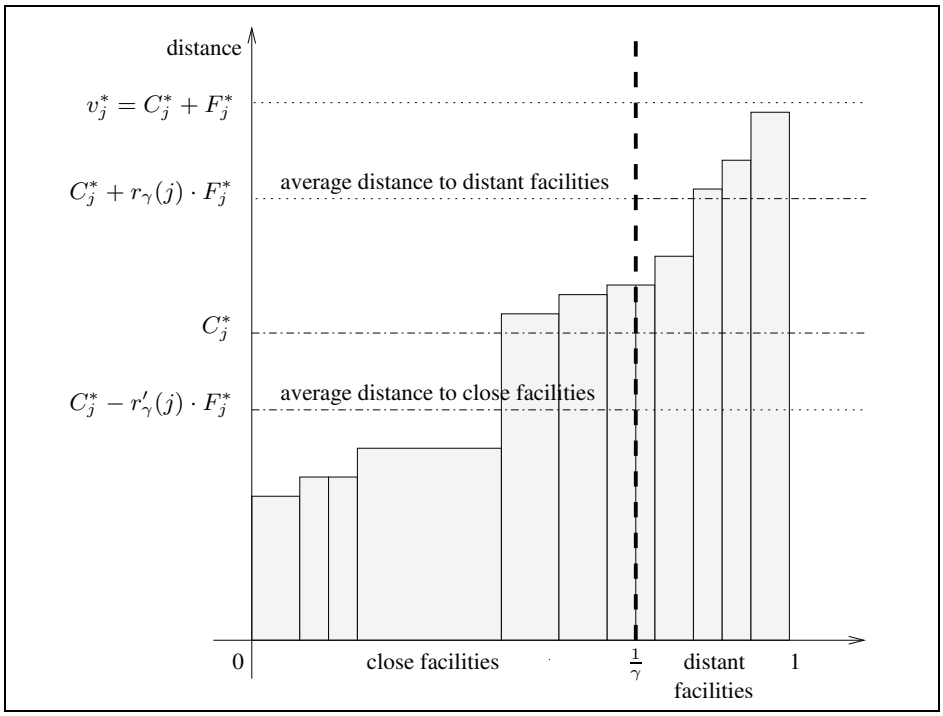

FIG. 3.1. *Distances to facilities serving client j; the width of a rectangle corresponding to facility i is equal to x_{ij}^*. The figure explains the meaning of $r_\gamma(j)$ and $r'_\gamma(j)$.*

The value $r_\gamma(j)$ is a measure of the irregularity of the instance around client j. It is the average distance to a distant facility minus the fractional connection cost C_j^* (note that $C_j^* = d(j, \mathcal{D}_j \cup \mathcal{C}_j)$ is the general average distance to both close and distant facilities) divided by the fractional facility cost of a client j; or it is equal to 0 if $F_j^* = 0$. Since $d(j, \mathcal{D}_j) \le v_j^*$, $C_j^* = d(j, \mathcal{D}_j \cup \mathcal{C}_j)$, and $C_j^* + F_j^* = v_j^*$, $r_\gamma(j)$ takes values between 0 and 1. $r_\gamma(j) = 0$ means that client j is served in the solution (x^*, y^*) by facilities that are all at the same distance. If $r_\gamma(j) = 1$, then the facilities are at different distances and the distant facilities are all so far from j that j is not willing to contribute to their opening. In fact, for clients j with $F_j^* = 0$ the value of $r_\gamma(j)$ is not relevant for our analysis.

Consider yet another quantity, namely, $r'_\gamma(j) = r_\gamma(j) \cdot (\gamma - 1)$. Observe that for a client j with $F_j^* > 0$ we have

$$r'_\gamma(j) = \frac{d(j, \mathcal{D}_j \cup \mathcal{C}_j) - d(j, \mathcal{C}_j)}{F_j^*}.$$

We may use the definitions of $r_\gamma(j)$ and $r'_\gamma(j)$ together with $C_j^* = d(j, \mathcal{D}_j \cup \mathcal{C}_c)$ to rewrite some distances from client j in the following form (see also Figure 3.1):

- the average distance to a close facility is

$$D_{av}^C(j) = d(j, \mathcal{C}_j) = C_j^* - r'_\gamma(j) \cdot F_j^*;$$

- the average distance to a distant facility is

$$D_{av}^D(j) = d(j, \mathcal{D}_j) = C_j^* + r_\gamma(j) \cdot F_j^*;$$

- the maximal distance to a close facility is

$$D^C_{max}(j) \le D^D_{av}(j) = C^*_j + r_\gamma(j) \cdot F^*_j.$$

In the following lemma we will prove an upper bound on the average distance from client j to another group of facilities.

LEMMA 3.3. *Suppose $\gamma < 2$ and that clients $j, j' \in \mathcal{C}$ are neighbors in $(\overline{x}, \overline{y})$, i.e., $\exists i \in \mathcal{F}$ s.t. $\overline{x}_{ij} > 0$ and $\overline{x}_{ij'} > 0$. Then either $\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j) = \emptyset$ or*

$$d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D^D_{av}(j) + D^C_{max}(j') + D^C_{av}(j').$$

Proof. Assume that $\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)$ is not empty, since otherwise we are done.

Case 1. Assume that the distance between j and j' is at most $D^D_{av}(j) + D^C_{av}(j')$. By a simple observation that a maximum is larger than the average, we get

$$d(j', \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D^C_{max}(j'). \tag{3.1}$$

Combining the assumption with (3.1), we obtain

$$d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D^D_{av}(j) + D^C_{max}(j') + D^C_{av}(j').$$

Case 2. Assume that the distance between j and j' is longer than $D^D_{av}(j) + D^C_{av}(j')$. Since $d(j, \mathcal{C}_j \cap \mathcal{C}_{j'}) \le D^D_{av}(j)$, the assumption implies

$$d(j', \mathcal{C}_j \cap \mathcal{C}_{j'}) > D^C_{av}(j'). \tag{3.2}$$

Consider the following two subcases.

Case 2a. Assume that $d(j', \mathcal{C}_{j'} \cap \mathcal{D}_j) \ge D^C_{av}(j')$. This assumption together with (3.2) gives

$$d(j', \mathcal{C}_{j'} \cap (\mathcal{C}_j \cup \mathcal{D}_j)) \ge D^C_{av}(j'). \tag{3.3}$$

Recall that $D^C_{av}(j') = d(j', \mathcal{C}_{j'})$. Hence (3.3) is equivalent to

$$d(j', \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D^C_{av}(j'). \tag{3.4}$$

Since j and j' are neighbors, the distance between them is at most $D^C_{max}(j) + D^C_{max}(j')$. By the triangle inequality (1.1) we may add this distance to (3.4) and get

$$d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D^D_{av}(j) + D^C_{max}(j') + D^C_{av}(j').$$

Case 2b. In the remaining case we assume that $d(j', \mathcal{C}_{j'} \cap \mathcal{D}_j) < D^C_{av}(j')$. This assumption may also be written as

$$d(j', \mathcal{C}_{j'} \cap \mathcal{D}_j) = D^C_{av}(j') - z \text{ for some } z > 0. \tag{3.5}$$

Now we combine (3.5) with the assumption of Case 2 to get

$$d(j, \mathcal{C}_{j'} \cap \mathcal{D}_j) \ge D^D_{av}(j) + z. \tag{3.6}$$

Let $\hat{y} = \sum_{i \in (\mathcal{C}_{j'} \cap \mathcal{D}_j)} \overline{y}_i$ be the total fractional opening of facilities in $\mathcal{C}_{j'} \cap \mathcal{D}_j$ in the modified fractional solution $(\overline{x}, \overline{y})$.

Observe that (3.6) together with the definition $d(j, \mathcal{D}_j) = D^D_{av}(j)$ implies that the set $(\mathcal{D}_j \setminus \mathcal{C}_{j'})$ is not empty. Moreover it contains facilities whose opening variables $\overline{y}$

sum up to $\gamma - 1 - \hat{y} > 0$. More precisely, inequality (3.6) implies $d(j, \mathcal{D}_j \setminus \mathcal{C}_{j'}) \leq D^D_{av}(j) - z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}$. Hence

$$D^C_{max}(j) \leq D^D_{av}(j) - z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}. \tag{3.7}$$

We combine (3.7) with the assumption of Case 2 to conclude that the minimal distance from j' to a facility in $\mathcal{C}_{j'} \cap \mathcal{C}_j$ is at least $D^D_{av}(j) + D^C_{av}(j') - D^C_{max}(j) \geq D^C_{av}(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}$. Hence

$$d(j', \mathcal{C}_{j'} \cap \mathcal{C}_j) \geq D^C_{av}(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}. \tag{3.8}$$

Recall that, by definition, $d(j', \mathcal{C}_{j'}) = D^C_{av}(j')$. Hence equality (3.5) may be written as

$$d(j', \mathcal{C}_{j'} \setminus \mathcal{D}_j) = D^C_{av}(j') + z \cdot \frac{\hat{y}}{1 - \hat{y}}. \tag{3.9}$$

Since, by the assumption that $\gamma < 2$, we have $\frac{\hat{y}}{1-\hat{y}} < \frac{\hat{y}}{\gamma - 1 - \hat{y}}$, we may also write

$$d(j', \mathcal{C}_{j'} \setminus \mathcal{D}_j) < D^C_{av}(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}. \tag{3.10}$$

We may now combine (3.10) with (3.8) to get

$$d(j', \mathcal{C}_{j'} \setminus (\mathcal{D}_j \cup \mathcal{C}_j)) < D^C_{av}(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}. \tag{3.11}$$

Finally, we bound the distance from j to j' by $D^C_{max}(j) + D^C_{max}(j')$ to get

$$\begin{aligned} d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) &\leq D^C_{max}(j) + D^C_{max}(j') + d(j', \mathcal{C}_{j'} \setminus (\mathcal{D}_j \cup \mathcal{C}_j)) \\ &\leq D^D_{av}(j) - z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}} + D^C_{max}(j') + D^C_{av}(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}} \\ &= D^D_{av}(j) + D^C_{max}(j') + D^D_{av}(j'), \end{aligned}$$

where the second inequality is an application of (3.11) and (3.7). ∎

4. Our new algorithm. Here we again state our algorithm (cf. section 1.3), but now we use the notation developed in the previous sections.

ALGORITHM $A1(\gamma)$.

1. Solve the LP-relaxation of the problem to obtain a solution (x^*, y^*).
2. Modify the fractional solution as described in section 3.2 to obtain a complete solution $(\overline{x}, \overline{y})$.
3. Compute a greedy clustering for the solution $(\overline{x}, \overline{y})$, choosing as cluster centers unclustered clients minimizing $D^C_{av}(j) + D^C_{max}(j)$.
4. For every cluster center j, open one of its close facilities randomly with probabilities $\overline{x}_{ij}$.
5. For each facility i that is not a close facility of any cluster center, open it independently with probability $\overline{y}_i$.
6. Connect each client to an open facility that is closest to it.

In order to upper bound the expected connection cost of the produced solution, we use the following special case of the FKG inequality (see Fortuin, Kasteleyn, and Ginibre [13]), which is a generalization of Chebyshev's sum inequality (see also [16]).

LEMMA 4.1. *Let $a, c, w \in (\{1, \dots, k\} \to \mathcal{R}^+)$ be k-element sequences, c nondecreasing and a nonincreasing. Then*

$$\frac{\sum_{i=1}^{k} a(i)w(i)c(i)}{\sum_{i=1}^{k} a(i)w(i)} \leq \frac{\sum_{i=1}^{k} w(i)c(i)}{\sum_{i=1}^{k} w(i)}.$$

The next lemma provides an upper bound on the expected distance from a client to the closest of the facilities opened by the algorithm within a certain subset of facilities. A version of this lemma was used in the analysis of most of the previous LP-rounding approximation algorithms for UFL.

LEMMA 4.2. *Let $y \in \{0,1\}^{|\mathcal{F}|}$ be a random binary vector encoding the facilities opened in steps 4 and 5 of Algorithm A1(γ); then the following inequality holds for any subset $A \subseteq \mathcal{F}$ of facilities, such that $\sum_{i \in A} \bar{y}_i > 0$, and any client $j \in \mathcal{C}$:*

$$E\left[\min_{i \in A,\, y_i = 1} c_{ij} \;\middle|\; \sum_{i \in A} y_i \geq 1\right] \leq d(j, A).$$

Proof. Observe that either the opening of facilities from A is pairwise independent or there exist disjoint subsets $A_1, A_2, \dots \subseteq A$, which correspond to clusters created in step 3 of the algorithm, such that the opening of facilities in each A_k is negatively correlated, but facilities from different sets are uncorrelated. The correlation in these subsets is a result of step 4 of the algorithm. In each such A_k, there is at most 1 facility opened, and the probability that one is opened equals $\sum_{i \in A_k} \overline{y}_i$.

For the purpose of this proof we analyze the following, possibly suboptimal, randomized process of assigning clients to open facilities. Consider an assignment algorithm that first creates a modified instance by replacing each A_k by a new facility i_k with distance to j equal to $d(j, A_k)$ and fractional opening $\overline{y}_{i_k} = \sum_{i \in A_k} \overline{y}_i$. The algorithm selects the closest open facility in the modified instance, which corresponds to choosing a facility or a set A_k in the original instance. In case a set A_k with an open facility is chosen, the algorithm selects the only open facility from the chosen set.

Obviously, the above-described algorithm may connect j to a facility that is not the closest of the open ones. Moreover, for certain instances, the expected connection cost to the facility chosen by this greedy algorithm is substantially higher than the expected connection cost to the closest open facility. Nevertheless, we will show that the greedy assignment of j to the closest open facility in the modified instance results in the expected connection cost at most equal to $d(j, A)$, which translates to the suboptimal assignment in the original instance, and therefore implies that in the optimal assignment in the original instance, the expected connection cost is also at most $d(j, A)$.

In the instance modified as described above, we have a set of facilities that are opened independently. It remains to prove the claim assuming independent opening of facilities, which we do with the help of Lemma 4.1. Consider the facilities from A in the order $i_1, i_2, \dots$ of nondecreasing distance from j. Since their opening is independent, the probability that i_l counts as the closest among the open facilities is

$$\begin{aligned} p_l &= \Pr[y_{i_1} = 0] \cdot \Pr[y_{i_2} = 0] \cdot \ldots \cdot \Pr[y_{i_{(l-1)}} = 0] \cdot \Pr[y_{i_l} = 1] \\ &= (1 - \overline{y}_{i_1})(1 - \overline{y}_{i_2}) \cdot \ldots \cdot (1 - \overline{y}_{i_{(l-1)}}) \cdot \overline{y}_{i_l}. \end{aligned}$$

The expected distance may be bounded as follows:

$$\begin{aligned} E\left[\min_{i\in A,\, y_i=1} c_{ij} \;\middle|\; \sum_{i\in A} y_i \geq 1\right] &= \frac{\sum_{l=1}^{|A|} p_l c_{i_l j}}{\sum_{l=1}^{|A|} p_l} \\ &= \frac{\sum_{l=1}^{|A|} \left(\Pi_{o=1}^{l-1}(1-\overline{y}_{i_o})\right) \overline{y}_{i_l} c_{i_l,j}}{\sum_{l=1}^{|A|} \left(\Pi_{o=1}^{l-1}(1-\overline{y}_{i_o})\right) \overline{y}_{i_l}} \\ &\leq \frac{\sum_{l=1}^{|A|} \overline{y}_{i_l} c_{i_l,j}}{\sum_{l=1}^{|A|} \overline{y}_{i_l}} \\ &= \frac{\sum_{i\in A} \overline{y}_{i_l} c_{i_l,j}}{\sum_{i\in A} \overline{y}_{i_l}} = d(j,A). \end{aligned}$$

The inequality in the above calculation is an application of Lemma 4.1 with $k = |A|$, $c(l) = c_{i_l,j}$, $w(l) = \overline{y}_{i_l}$, and $a(l) = \Pi_{o=1}^{l-1}(1-\overline{y}_{i_o})$. ∎

In the analysis of our algorithm we will also use the following result.

LEMMA 4.3. *Given are n independent events that occur with probabilities $p_1, p_2, \ldots, p_n$, respectively. The probability that at least one of these events occurs is at least equal to $1 - \frac{1}{e^{\sum_{i=1}^{n} p_i}}$, where e denotes the base of the natural logarithm.*

Let γ_0 be defined as the only positive solution to the equation

$$\frac{1}{e} + \frac{1}{e^{\gamma_0}} - (\gamma_0 - 1)\cdot\left(1 - \frac{1}{e} + \frac{1}{e^{\gamma_0}}\right) = 0. \tag{4.1}$$

An approximate value of this constant is $\gamma_0 \approx 1.67736$. As we will observe in the proof of Theorem 4.4, equation (4.1) appears naturally in the analysis of Algorithm $A1(\gamma)$.

THEOREM 4.4. *Algorithm $A1(\gamma_0)$ produces a solution with expected cost*

$$E[cost(SOL)] \leq \gamma_0 \cdot F^* + 1 + \frac{2}{e^{\gamma_0}} \cdot C^*.$$

Proof. The expected facility opening cost of the solution is

$$E[F_{SOL}] = \sum_{i\in\mathcal{F}} f_i \overline{y}_i \gamma \cdot \sum_{i\in\mathcal{F}} f_i y_i^* = \gamma \cdot F^*.$$

To bound the expected connection cost we show that for each client j there is, with a certain probability, an open facility within a certain distance. If j is a cluster center, one of its close facilities is open, and the expected distance to this open facility is $D_{av}^C(j) = C_j^* - r'_\gamma(j) \cdot F_j^* \leq C_j^*$.

If j is not a cluster center, it first considers its close facilities (see Figure 4.1). If any of them is open, by Lemma 4.2 the expected distance to the closest open facility is at most $D_{av}^C(j)$. From Lemma 4.3, at least one close facility is open with probability $p_c \geq (1 - \frac{1}{e})$.

Suppose none of the close facilities of j is open, but at least one of its distant facilities is open. Let p_d denote the probability of this event. Again by Lemma 4.2, the expected distance to the closest facility is then at most $D_{av}^D(j)$.

If neither any close nor any distant facility of client j is open, then j may connect itself to the facility serving its cluster center j'. Again from Lemma 4.3, such an event happens with probability $p_s \leq \frac{1}{e^\gamma}$. We will now use the fact that if $\gamma < 2$, then, by

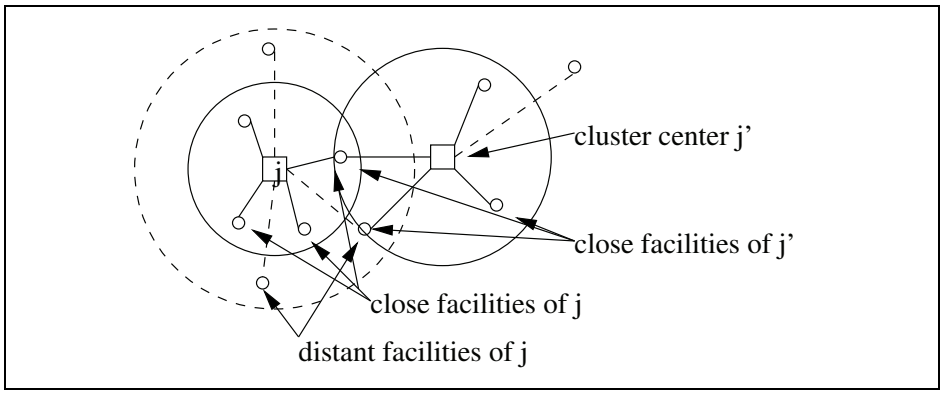

FIG. 4.1. *Facilities that client j may consider: its close facilities, distant facilities, and close facilities of cluster center j'.*

Lemmas 3.3 and 4.2, the expected distance from j to the facility opened around j' is at most $D^D_{av}(j) + D^C_{max}(j') + D^C_{av}(j')$.

Finally, we combine the probabilities of particular cases with the bounds on the expected connection for each of the cases to obtain the following upper bound on the expected total connection cost:

$$\begin{aligned}
E[C_{SOL}] &\le \sum_{j\in\mathcal{C}} \left(p_c \cdot D^C_{av}(j) + p_d \cdot D^D_{av}(j) + p_s \cdot (D^D_{av}(j) + D^C_{max}(j') + D^C_{av}(j'))\right) \\
&\le \sum_{j\in\mathcal{C}} \left((p_c + p_s) \cdot D^C_{av}(j) + (p_d + 2p_s) \cdot D^D_{av}(j)\right) \\
&= \sum_{j\in\mathcal{C}} \left((p_c + p_s) \cdot (C^*_j - r'_\gamma(j) \cdot F^*_j) + (p_d + 2p_s) \cdot (C^*_j + r_\gamma(j) \cdot F^*_j)\right) \\
&= ((p_c + p_d + p_s) + 2p_s) \cdot C^* \\
&\quad + \sum_{j\in\mathcal{C}} \left((p_c + p_s) \cdot (-r_\gamma(j) \cdot (\gamma - 1) \cdot F^*_j) + (p_d + 2p_s) \cdot (r_\gamma(j) \cdot F^*_j)\right) \\
&= (1 + 2p_s) \cdot C^* + \sum_{j\in\mathcal{C}} \left(F^*_j \cdot r_\gamma(j) \cdot (p_d + 2p_s - (\gamma - 1) \cdot (p_c + p_s))\right) \\
&\le \left(1 + \frac{2}{e^\gamma}\right) \cdot C^* + \sum_{j\in\mathcal{C}} \left(F^*_j \cdot r_\gamma(j) \cdot \left(\frac{1}{e} + \frac{1}{e^\gamma} - (\gamma - 1) \cdot \left(1 - \frac{1}{e} + \frac{1}{e^\gamma}\right)\right)\right).
\end{aligned}$$

In the above calculation we used the following properties. In the first inequality we explored the fact that cluster centers were chosen greedily, which implies $D^C_{max}(j') + D^C_{av}(j') \le D^C_{max}(j) + D^C_{av}(j)$. For the last inequality, we used $p_d + 2p_s = 1 - p_c + p_s \le 1 - (1 - \frac{1}{e}) + \frac{1}{e^\gamma} = \frac{1}{e} + \frac{1}{e^\gamma}$.

It remains to observe that by setting $\gamma = \gamma_0 \approx 1.67736$ (see (4.1)) we eliminate the last term in the connection cost bound, and we obtain $E[C_{SOL}] \le (1 + \frac{2}{e^{\gamma_0}}) \cdot C^* \le 1.37374 \cdot C^*$ (see Figure 4.2). ∎

Algorithm $A1(\gamma_0)$ was described as a procedure of rounding a particular fractional solution to the LP-relaxation of the problem. In the presented analysis we compared the cost of the obtained solution with the cost of the starting fractional solution. If we appropriately scale the cost function in the LP-relaxation before solving the relaxation, we easily obtain an algorithm with a bifactor approximation guarantee in

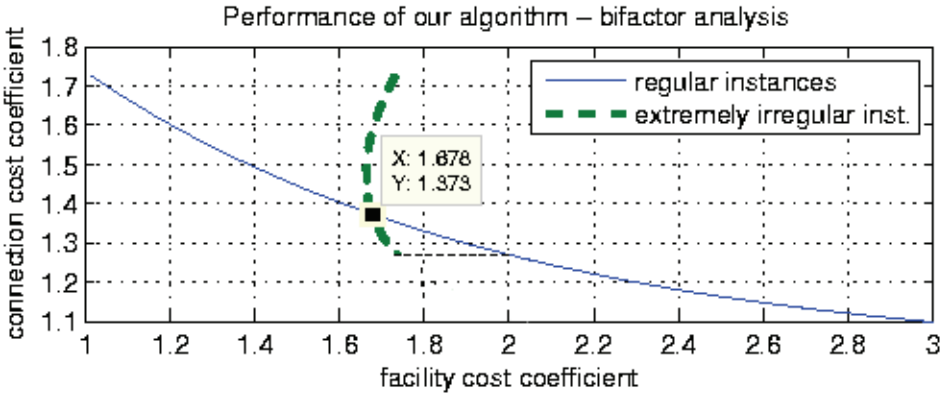

FIG. 4.2. *The performance of our algorithm for different values of parameter γ. The solid line corresponds to regular instances with $r_\gamma(j) = 0$ for all j, and it coincides with the approximability lower bound curve. The dashed line corresponds to instances with $r_\gamma(j) = 1$ for all j. For a particular choice of γ we get a horizontal segment connecting those two curves; for $\gamma \approx 1.67736$ the segment becomes a single point. Observe that for instances dominated by connection cost, only a regular instance may be tight for the lower bound.*

a stronger sense. Namely, we get a comparison of the produced solution with any feasible solution to the LP-relaxation of the problem. Such a stronger guarantee is, however, not necessary to construct the 1.5-approximation algorithm for the metric UFL problem, which is presented in the next section.

Algorithm $A1(\gamma)$ with $\gamma = 1 + \epsilon$ (for a sufficiently small positive ϵ) is essentially the algorithm of Chudak and Shmoys. Observe that for regular instances, namely, those with $r_\gamma(j) = 0$ for every client j, we do not need to set $\gamma = \gamma_0$ to eliminate the dependence of connection cost of the produced solution on the facility opening cost of the fractional solution. Hence, for regular instances, we get a $(\gamma, \frac{2}{e^\gamma})$-approximation algorithm for each choice of $\gamma > 1$.

5. The 1.5-approximation algorithm. In this section we will combine our algorithm with an earlier algorithm of Jain et al. to obtain a 1.5-approximation algorithm for the metric UFL problem.

In 2002 Jain, Mahdian, and Saberi [18] proposed a primal-dual approximation algorithm (the JMS algorithm). Using a dual fitting approach they showed that it is a 1.61-approximation algorithm. Later Mahdian, Ye, and Zhang [20] derived the following result.

LEMMA 5.1 (see [20]). *The cost of a solution produced by the JMS algorithm is at most $1.11 \times F^* + 1.7764 \times C^*$, where F^* and C^* are facility and connection costs in an optimal solution to the linear relaxation of the problem.*

THEOREM 5.2. *Consider the solutions obtained with the $A1(\gamma_0)$ and JMS algorithms. The cheapest solution of the two is expected to have a cost at most 1.5 times the cost of the optimal fractional solution.*

Proof. Consider an algorithm $A2$ that does the following. With probability $p = 0.313$ runs the JMS algorithm, and otherwise, with probability $1 - p$, runs the $A1(\gamma_0)$ algorithm. Suppose that we are given an instance, and that F^* and C^* are facility and connection costs in an optimal solution to the linear relaxation of this instance. Consider the expected cost of the solution produced by Algorithm $A2$ for this instance: $E[cost] \le p \cdot (1.11 \cdot F^* + 1.7764 \cdot C^*) + (1 - p) \cdot (1.67736 \cdot F^* + 1.37374 \cdot C^*) =$

$1.4998 \cdot F^* + 1.4998 \cdot C^* < 1.5 * (F^* + C^*) \leq 1.5 * OPT.$ □

Instead of the JMS algorithm we could take the algorithm of Mahdian et al. [20], the MYZ(δ) algorithm, that scales the facility costs by δ, runs the JMS algorithms, scales back the facility costs, and finally runs the greedy augmentation procedure. With the notation introduced in section 2.3, the MYZ(δ) algorithm is the S_δ(JMS) algorithm. The MYZ(1.504) algorithm was proven [20] to be a 1.52-approximation algorithm for the metric UFL problem. We may change the value of δ in the original analysis to observe that MYZ(1.1) is a $(1.2053, 1.7058)$-approximation algorithm. This algorithm combined with our $A1(\gamma_0)$ $(1.67736, 1.37374)$-approximation algorithm gives a 1.4991-approximation algorithm for UFL. This shows how much improvement we obtain by using the scaling technique on the greedy algorithm's side.

6. Multilevel facility location. In the k-level facility location problem the clients need to be connected to open facilities on the first level, and each open facility except on the last, kth level needs to be connected to an open facility on the next level. Aardal, Chudak, and Shmoys [1] gave a 3-approximation algorithm for the k-level problem with arbitrary k. Ageev, Ye, and Zhang [3] proposed a reduction of a k-level problem to a $(k-1)$-level and a 1-level problem, which results in a recursive algorithm. This algorithm uses an approximation algorithm for the single-level problem and has a better approximation ratio, but only for instances with small k. Using our new Algorithm $A1(\gamma_0)$ instead of the JMS algorithm within this framework improves approximation for each level. In particular, in the limit as k tends to ∞, we get a 3.236-approximation, which is the best possible for this construction.

By a slightly different method, Zhang [26] obtained a 1.77-approximation algorithm for the 2-level problem. For the 3-level and the 4-level versions of the problem he obtained 2.523-[2] and 2.81-approximation algorithms by reducing to a problem with a smaller number of levels. In the following section we will modify the algorithm by Zhang for the 3-level problem and use the new $(1.67736, 1.37374)$-approximation algorithm for the single-level part, to obtain a 2.492-approximation, which improves on the previously best known approximation by Zhang. Note that for $k > 4$ the best known approximation factor is still due to Aardal et al. [1].

6.1. 3-level facility location. We will now present the ingredients of the 2.492-approximation algorithm. We start from an algorithm to solve the 2-level version.

LEMMA 6.1 (Theorem 2 in [26]). *The 2-level UFL problem may be approximated by a factor of* $1.77 + \epsilon$ *in polynomial time for any given constant* $\epsilon > 0$.

Zhang [26] also considered a scaling technique analogous to the one described in section 2.3, but applicable to the 2-level version of the problem. An effect of using this technique is analyzed in the following lemma.

LEMMA 6.2 (Theorem 3 in [26]). *For any given* $\epsilon > 0$, *if there is an* (a, b)*-approximation algorithm for the 2-level UFL problem, then we can obtain an approximation algorithm for the 2-level UFL problem with performance guarantee*

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

for any $\Delta \geq 1$.

He also uses the following reduction.

[2]This value deviates slightly from the value 2.51 given in the paper. The original argument contained a minor calculation error.

LEMMA 6.3 (Lemma 7 in [26]). *Assume that the* 1*-level and* 2*-level UFL problems have approximation algorithms with factors* (a, b) *and* (α, β)*, respectively; then the* 3*-level UFL problem may be approximated by factors* $(\max\{a, \frac{a+\alpha}{2}\}, \frac{3b+\beta}{2})$.

Zhang [26] observed that the above three statements may be combined with the MYZ algorithm to improve the approximation ratio for the 3-level UFL problem. In the following theorem we show that we may use our new $(1.6774, 1.3738)$-approximation algorithm for the 1-level UFL problem to get an even better approximation for the 3-level variant.

THEOREM 6.4. *There is a* 2.492*-approximation algorithm for the* 3*-level UFL problem.*

Proof. We first use the algorithm from Lemma 6.1, and the scaling technique from Lemma 6.2, with $\Delta = 1.57971$, to obtain a $(2.492, 1.48743)$-approximation algorithm for the 2-level UFL problem.

Then we use our $(1.6774, 1.3738)$-approximation algorithm for the 1-level UFL problem with the scaling technique from Lemma 2.1, with $\gamma = 2.25827$, to obtain a $(2.492, 1.1655)$-approximation algorithm for the 1-level UFL problem.

Finally, we use Lemma 6.3 to combine these two algorithms into a $(2.492, 2.492)$-approximation algorithm for the 3-level UFL problem. ∎

7. Universal randomized clustering procedure. In this section we discuss a different approach to clustering. We propose to modify the greedy clustering algorithm by choosing consecutive cluster centers randomly with uniform distribution. The output of such a process is obviously random, but we may still prove some statements about probabilities. A resulting clustering will be denoted by a function $g : \mathcal{C} \to \mathcal{C}$, that assigns to each client j the center of its cluster $j' = g(j)$. The following lemma states that the clustering g obtained with the randomized clustering procedure is expected to be "fair."

LEMMA 7.1. *Given a graph* $G = (\mathcal{F} \cup \mathcal{C}, E)$ *and assuming that a clustering* g *was obtained by the above described random process, for every two distinct clients* j *and* j'*, the probability that* $g(j) = j'$ *is equal to the probability that* $g(j') = j$.

Proof. Let $C(G)$ denote the maximal (over the possible random choices of the algorithm) number of clusters that can be obtained from G with the random clustering procedure. The proof will be by induction on $C(G)$. Fix any $j, j' \in \mathcal{C}$ such that j is a neighbor of j' in G (if they are not neighbors, neither $g(j) = j'$ nor $g(j') = j$ can occur). Suppose $C(G) = 1$, then $\Pr[g(j) = j'] = \Pr[g(j') = j] = 1/|\mathcal{C}|$.

Let us now assume that $C(G) > 1$. There are two possibilities: either one of j, j' will belong to the first cluster or none of them will. Consider the first case (the first chosen cluster center is either j or j' or one of their neighbors). If j (j') is chosen as a cluster center, then $g(j') = j$ ($g(j) = j'$). Since they are chosen with the same probability, the contribution of the first case to the probability of $g(j') = j$ is equal to the contribution to the probability of $g(j) = j'$. If neither of them gets chosen as a cluster center but at least one belongs to the new cluster, then neither $g(j') = j$ nor $g(j) = j'$ is possible.

Now consider the second case (neither j nor j' belongs to the first cluster). Consider the graph G' obtained from G by removing the first cluster. The random clustering proceeds like it has just started with the graph G', but the maximal number of possible clusters is smaller: $C(G') \leq C(G) - 1$. Therefore, by the inductive hypothesis, in a random clustering of G' the probability that $g(j') = j$ is equal to the probability that $g(j) = j'$. ∎

If $g(j) = j'$ in a clustering g of graph G, we will say that client j' *offers support*

to client j. The main idea behind the clustering algorithms for the UFL problem is that we may afford to serve each cluster center directly (because they are never neighbors in G) and all the other clients are offered support from their cluster centers. A noncentral client may either accept the support and connect itself via its cluster center (that is what all noncentral clients do in the algorithm of Shmoys et al.), or it may try to get served locally, and if it fails, accept the support (this is the way the Chudak and Shmoys algorithm works). In both algorithms the probability that an offer of support is accepted is estimated to be constant. Therefore, we may modify those algorithms to use the random clustering procedure and do the following analysis.

For any two clients j and j', the probability that j accepts the support of j' is equal to the probability that j' accepts the support of j. Let i be a facility on a shortest path from j to j'. When we compute the expected connection cost of client j, we observe that with certain probability p it accepts the support of j'. In such a case it must pay for the route via i and j' to the facility directly serving j'. We will now change the bookkeeping and say that in this situation j is paying only for the part until facility i, and the rest is paid by j', but if j would be supporting j' it would have to pay a part of j''s connection cost, which is the length of the path from i via j to the facility serving j. We may think of this as each client having a bank account, and when it accepts support it makes a deposit, and when it offers support and the support is accepted, then it withdraws money to pay a part of the connection cost of the supported client. From Lemma 7.1 we know that for a client j the probability that it will earn on j' is equal to the probability that it will lose on j'. Therefore, if the deposited amount is equal to the withdrawal, the expected net cash flow is zero.

The above analysis shows that randomizing the clustering phase of the known LP-rounding algorithms would not worsen their approximation ratios. Although it does not make much sense to use a randomized algorithm if it has no better performance guarantee, the random clustering has an advantage of allowing the analysis to be more local and uniform.

8. Concluding remarks. With the 1.52-approximation algorithm of Mahdian et al. it was not clear to the authors if a better analysis of the algorithm could close the gap with the approximation lower bound of 1.463 by Guha and Khuller. In [6] we have recently given a negative answer to this question by constructing instances that are hard for the MYZ algorithm. Similarly, we now do not know if our new Algorithm $A1(\gamma)$ could be analyzed better to close the gap. Construction of hard instances for our algorithm remains an open problem.

The technique described in section 2.3 enables us to move the bifactor approximation guarantee of an algorithm along the approximability lower bound of Jain et al. (see Figure 1.1) towards higher facility opening costs. If we developed a technique to move the analysis in the opposite direction, together with our new algorithm, it would imply closing the approximability gap for the metric UFL problem. It seems that with such an approach we would have to face the difficulty of analyzing an algorithm that closes some of the previously opened facilities.

Acknowledgments. The authors want to thank David Shmoys, Steven Kelk, Evangelos Markakis, Andreas Karrenbauer, and the anonymous referees of this submission as well as the referees of the earlier conference version of this paper [5] for their advice and valuable remarks.

REFERENCES

[1] K. Aardal, F. Chudak, and D. B. Shmoys, *A 3-approximation algorithm for the k-level uncapacitated facility location problem*, Inform. Process. Lett., 72 (1999), pp. 161–167.

[2] A. A. Ageev and M. I. Sviridenko, *A 0.828-approximation algorithm for the uncapacitated facility location problem*, Discrete Appl. Math., 93 (1999), pp. 149–156.

[3] A. Ageev, Y. Ye, and J. Zhang, *Improved combinatorial approximation algorithms for the k-level facility location problem*, in Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 2719, Springer, Berlin, 2003, pp. 145–156.

[4] S. Arora, P. Raghavan, and S. Rao, *Approximation schemes for Euclidean k-medians and related problems*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 1998, pp. 106–113.

[5] J. Byrka, *An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem*, in Proceedings of the 10th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Lecture Note in Comput. Sci. 4627, Springer, Berlin, 2007, pp. 29–43.

[6] J. Byrka and K. Aardal, *The approximation gap for the metric facility location problem is not yet closed*, Oper. Res. Lett., 35 (2007), pp. 379–384.

[7] M. Charikar and S. Guha, *Improved combinatorial algorithms for facility location and k-median problems*, in Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS), 1999, pp. 378–388.

[8] M. Charikar and S. Guha, *Improved combinatorial algorithms for facility location problems*, SIAM J. Comput., 34 (2005), pp. 803–824.

[9] F. A. Chudak, *Improved approximation algorithms for uncapacitated facility location*, in Proceedings of the 6th Integer Programing and Combinatorial Optimization (IPCO), Lecture Note in Comput. Sci. 1412, Springer, Berlin, 1998, pp. 180–194.

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

[11] G. Cornuéjols, M. L. Fisher, and G. L. Nemhauser, *Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms*, Management Sci., 8 (1977), pp. 789–810.

[12] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[13] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre, *Correlation inequalities on some partially ordered sets*, Comm. Math. Phys., 22 (1971), pp. 89–103.

[14] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, in Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 1998, pp. 649–657.

[15] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms, 31 (1999), pp. 228–248.

[16] M. Hazewinkel, ed., *Encyclopedia of Mathematics*, Springer, Berlin, 2002, http://eom.springer.de/default.htm.

[17] D. S. Hochbaum, *Heuristics for the fixed cost median problem*, Math. Programming, 22 (1982), pp. 148–162.

[18] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 2002, pp. 731–740.

[19] J.-H. Lin and J. S. Vitter, *ϵ-approximations with minimum packing constraint violation*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 1992, pp. 771–782.

[20] M. Mahdian, Y. Ye, and J. Zhang, *Improved approximation algorithms for metric facility location problems*, in Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Lecture Note in Comput. Sci. 2462, Springer, Berlin, 2002, pp. 229–242.

[21] M. Mahdian, Y. Ye, and J. Zhang, *Approximation algorithms for metric facility location problems*, SIAM J. Comput., 36 (2006), pp. 411–432.

[22] D. B. Shmoys, *Approximation algorithms for facility location problems*, in Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Lecture Notes in Comput. Sci. 1913, Springer, Berlin, 2000, pp. 265–274.

[23] D. B. Shmoys, É. Tardos, and K. Aardal, *Approximation algorithms for facility location problems (extended abstract)*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 1997, pp. 265–274.

[24] M. Sviridenko, *An improved approximation algorithm for the metric uncapacitated facility location problem*, in Proceedings of the 9th Integer Programming and Combinatorial Optimization (IPCO), Lecture Note in Comput. Sci. 2337, Springer, Berlin, 2002, pp. 240–257.

[25] J. Vygen, *Approximation Algorithms for Facility Location Problems (Lecture Notes)*, Report 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005; available online from http://www.or.uni-bonn.de/~vygen/fl.pdf.

[26] J. Zhang, *Approximating the two-level facility location problem via a quasi-greedy approach*, Math. Program., 108 (2006), pp. 159–176.