An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem

Jaroslaw Byrka[1]

Centrum voor Wiskunde en Informatica,
Kruislaan 413, NL-1098 SJ Amsterdam, Netherlands
J.Byrka@cwi.nl

Abstract. We consider the metric uncapacitated facility location problem(UFL). In this paper we modify the $(1 + 2/e)$-approximation algorithm of Chudak and Shmoys to obtain a new (1.6774,1.3738)-approximation algorithm for the UFL problem. Our linear programing rounding algorithm is the first one that touches the approximability limit curve $(\gamma_f, 1 + 2e^{-\gamma_f})$ established by Jain et al. As a consequence, we obtain the first optimal approximation algorithm for instances dominated by connection costs.
Our new algorithm - when combined with a (1.11,1.7764)-approximation algorithm proposed by Jain, Mahdian and Saberi, and later analyzed by Mahdian, Ye and Zhang - gives a 1.5-approximation algorithm for the metric UFL problem. This algorithm improves over the previously best known 1.52-approximation algorithm by Mahdian, Ye and Zhang, and it cuts the gap with the approximability lower bound by 1/3.
The algorithm is also used to improve the approximation ratio for the 3-level version of the problem.

1 Introduction

The Uncapacitated Facility Location (UFL) problem is defined as follows. We are given a set $\mathcal{F}$ of n_f *facilities* and a set $\mathcal{C}$ of n_c *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 [8]). A UFL instance is *metric* if its *connection cost* function satisfies a kind of *triangle inequality*, namely if $c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$ for any $i, i' \in \mathcal{C}$ and $j, j' \in \mathcal{F}$.

The UFL problem has a rich history starting in the 1960's. The first results on approximation algorithms are due to Cornuéjols, Fisher, and Nemhauser [7] who considered the problem with an objective function of maximizing the "profit" of

[1] Supported by the EU Marie Curie Research Training Network ADONET, Contract No MRTN-CT-2003-504438

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. For the objective function of minimizing the sum of connection cost and opening cost, Hochbaum [9] presented a greedy algorithm with an $O(\log n)$ approximation guarantee, where n is the number of clients. The first approximation algorithm with constant approximation ratio for the minimization problem where the connection costs satisfy the triangle inequality, was developed by Shmoys, Tardos, and Aardal [14]. Several approximation algorithms have been proposed for the metric UFL problem after that, see for instance [8, 4–6, 15, 10, 12]. Up to now, the best known approximation ratio was 1.52, obtained by Mahdian, Ye, and Zhang [12]. Many more algorithms have been considered for the UFL problem and its variants. We refer an interested reader to survey papers by Shmoys [13] and Vygen [16].

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* studied by Charikar and Guha [4]. We say that an algorithm is a (λ_f,λ_c)-approximation algorithm if the solution it delivers has total 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.

Guha and Khuller [8] 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 [16]). Jain et al. [10] 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 $NP \subseteq DTIME(n^{\log \log n})$.

1.1 Our contribution

We modify the $(1+2/e)$-approximation algorithm of Chudak [5], see also Chudak and Shmoys [6], to obtain a new (1.6774,1.3738)-approximation algorithm for the UFL problem. Our linear programing (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 et al. 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).

Our main technique is to modify the support graph corresponding to the LP solution before clustering, and to use various average distances in the fractional solution to bound the cost of the obtained solution. Modifying the solution in such a way was introduced by Lin and Vitter [11] and is called *filtering*. Throughout this paper we will use the name *sparsening technique* for the combination of filtering with our new analysis.

One could view our contribution as an improved analysis of a minor modification of the algorithm by Sviridenko [15], which also introduces filtering to the

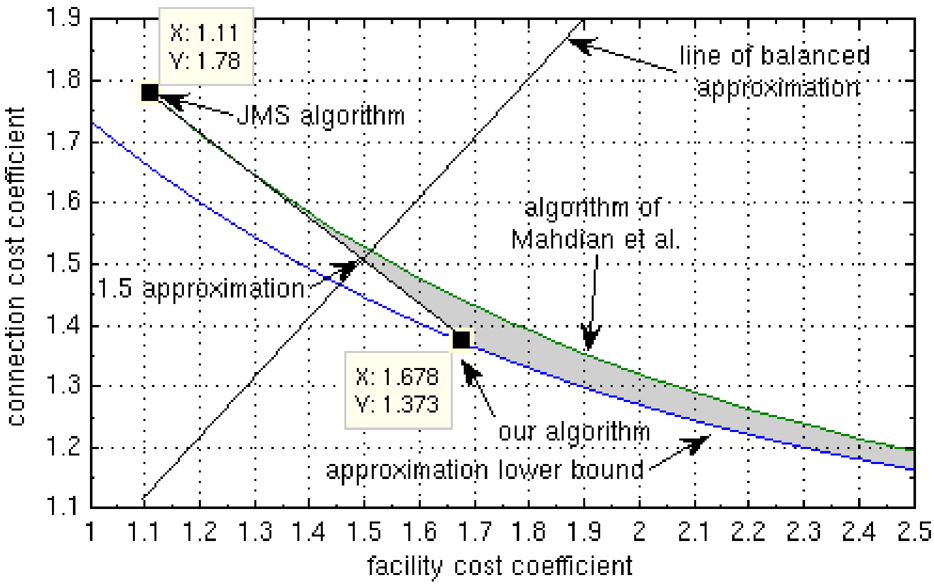

Fig. 1. Bifactor approximation picture. The gray area corresponds to the improvement due to our algorithm.

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 do describe and analyze.

The motivation for the sparsening technique is the "irregularity" of instances that are potentially tight for the original algorithm of Chudak and Shmoys. We propose a way of measuring and controlling this irregularity. In fact our clustering is the same as the one used by Sviridenko in his 1.58-approximation algorithm [15], but we continue our algorithm in the spirit of Chudak and Shmoys' algorithm, which leads to an improved bifactor approximation guaranty.

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 of a 1/3 of the gap with the approximation lower bound by Guha and Khuler [8].

We also note that the new (1.6774,1.3738)-approximation algorithm may be used to improve the approximation ratio for the 3-level version of the UFL problem to 2.492.

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 programing (IP) formulation of the problem, and then round the fractional solution to produce an integral 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 integer programming problem.

$$\begin{array}{lll} \text{minimize} & \sum_{i\in\mathcal{F},j\in\mathcal{C}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i & \\ & & \\ \text{subject to} & \sum_{i\in\mathcal{F}} x_{ij} = 1 & \text{for all } j \in \mathcal{C} \quad (1) \\ & x_{ij} - y_i \leq 0 & \text{for all } i \in \mathcal{F}, j \in \mathcal{C} \ (2) \\ & x_{ij}, y_i \in \{0,1\} & \text{for all } i \in \mathcal{F}, j \in \mathcal{C} \ (3) \end{array}$$

A linear relaxation of this IP formulation is obtained by replacing Condition (3) by the condition $x_{ij} \geq 0$ 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.

$$\begin{array}{lll} \text{maximize} & \sum_{j\in\mathcal{C}} v_j & \\ & & \\ \text{subject to} & \sum_{j\in\mathcal{C}} w_{ij} \leq f_i & \text{for all } i \in \mathcal{F} \quad (4) \\ & v_j - w_{ij} \leq c_{ij} & \text{for all } i \in \mathcal{F}, j \in \mathcal{C} \ (5) \\ & w_{ij} \geq 0 & \text{for all } i \in \mathcal{F}, j \in \mathcal{C} \ (6) \end{array}$$

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 with vertices being the facilities and the clients of the instance, and where there is an edge between a client j and a facility i 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.

The clustering of 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

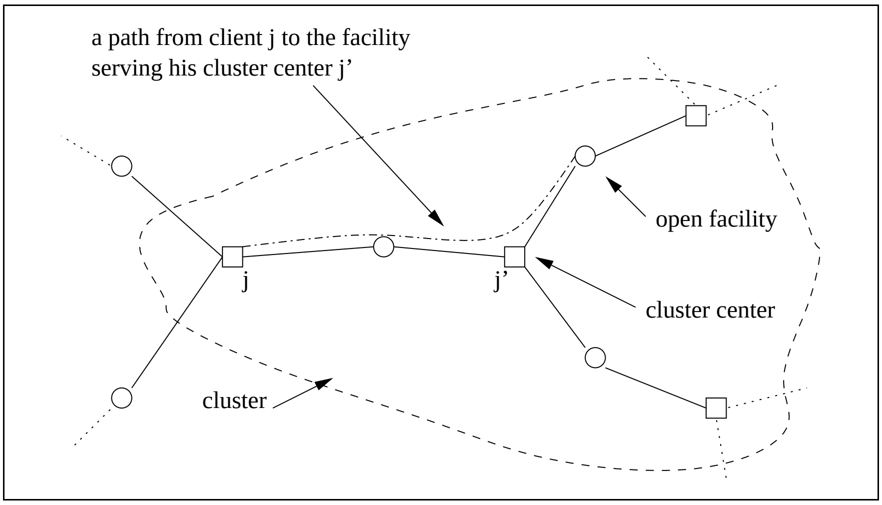

Fig. 2. A cluster. If we make sure that at least one facility is open around a cluster center j' , then any other client j from the cluster may use this facility. Because the connection costs are assumed to be metric, the distance to this facility is at most the length of the shortest path from j to the open facility.

are neighbors in the support graph. This property helps us to open one of the adjacent facilities for each cluster center. Formally we will say that a clustering is a function $g : \mathcal{C} \rightarrow \mathcal{C}$ that assigns each client to the center of his cluster. For a picture of a cluster see Figure 2.

All the above mentioned algorithms 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 [15].

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 [8] (see also [4]) is the following. 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 impact of opening facility i on the total cost of the solution, also called the *gain* of opening i, denoted by g_i. The greedy augmentation procedure, while there is a facility i with positive gain g_i, opens a facility i_0 that maximizes the ratio of saved cost to the facility opening cost $\frac{g_i}{f_i}$, and updates values of g_i. The procedure terminates when there is no facility whose opening would decrease the total cost.

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

1. scale up all facility opening costs by a factor 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 [12] one may prove the following lemma.

Lemma 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 an (λ_f,λ_c)-approximation algorithm with $\lambda_f << \lambda_c$. 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 solution. It is based on modifying a fractional solution in a way introduced by Lin and Vitter [11] and called *filtering*.

The filtering technique has been successfully applied to the facility location problem, also in the algorithms of Shmoys, Tardos, and Aardal [14] and of Sviridenko [15]. We will give an alternative analysis of what is the effect of applying 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 we have an optimal primal solution (x^*, y^*) and the corresponding optimal dual solution (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 his 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 some irregularities of an instance if they occur. We call an instance *regular* if the facilities that fractionally serve a client j are all at the same distance from j. 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 also follows from our analysis in Section 4. It remains to use the technique described in section 2.3 to obtain an optimal $1.463\ldots$-approximation algorithm for such regular instances.

The instances that are not regular are called *irregular*. Difficult to understand are the irregular instances. In fractional solutions for these instances particular clients 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. We will remove links to distant facilities before the clustering step, so that if there are irregularities, distances to cluster centers should decrease.

We measure the local irregularity of an instance by comparing a fractional connection cost of a client to the average distance to his distant facilities. In the case of a regular instance, the sparsening technique gives the same results as technique described in section 2.3, but for irregular instances sparsening also takes some advantage of the irregularity.

3.2 Details

We will start by modifying the primal optimal fractional solution (x^*, y^*) by scaling the y-variables by a constant $\gamma > 1$ to obtain a suboptimal fractional solution $(x^*, \gamma \cdot y^*)$. Now suppose that the y-variables are fixed, but that we now have a freedom to change the x-variables in order to minimize the total cost. For each client j we change the corresponding x-variables so that he uses his closest facilities 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 facilities fractionally serving j only the latest one in the chosen ordering may be opened more then it serves j. Formally, for any facilities i and i' such that i' is later in the ordering, if $x_{ij} < y_i$ then $x_{i'j} = 0$.

Without loss of generality, we may assume that this solution is complete (i.e. there are no $i \in \mathcal{F}, j \in \mathcal{C}$ such that $0 < x_{ij} < y_i$). Otherwise we may split facilities to obtain an equivalent instance with a complete solution - see [15][Lemma 1] for a more detailed argument.

Let $(\overline{x}, \overline{y})$ denote the obtained complete solution. For a client j we say that a facility i is one of *his close facilities* if it fractionally serves client j in $(\overline{x}, \overline{y})$. 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.

Definition 1. *Let*

$$r_\gamma(j) = \begin{cases} \dfrac{\frac{\gamma}{\gamma-1}\sum_{i \in \{i \in \mathcal{F} | \overline{x}_{ij}=0\}} c_{ij} x^*_{ij} - C^*_j}{F^*_j} & \text{for } F^*_j > 0 \\ 0 & \text{for } F^*_j = 0. \end{cases}$$

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 (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 0 if $F^*_j = 0$.

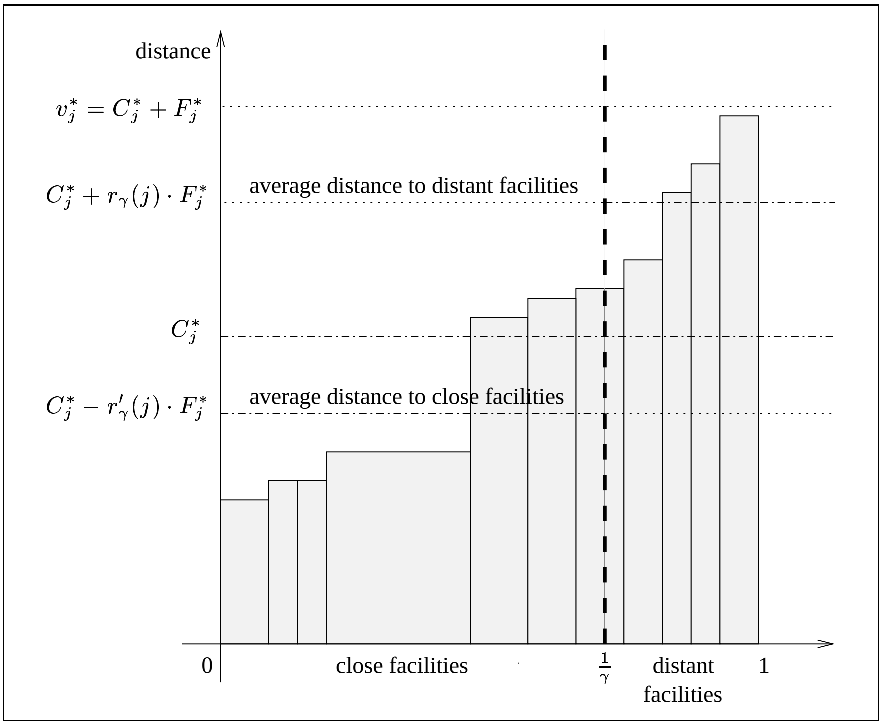

Fig. 3. Distances to facilities serving client j; the width of a rectangle corresponding to facility i is equal to x_{ij}^*. Figure explains the meaning of $r_\gamma(j)$.

Observe, that $r_\gamma(j)$ 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. In the case of $r_\gamma(j) = 1$ 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.

To get some more intuition for the F_j^* and $r_\gamma(j)$ values, imagine that you know F_j^* and C_j^*, but the adversary is constructing the fractional solution and he decided about distances to particular facilities fractionally serving client j. One could interpret F_j^* as a measure of freedom the adversary has when he chooses those distances. In this language, $r_\gamma(j)$ is a measure of what fraction of this freedom is used to make distant facilities more distant than average facilities.

Let $r'_\gamma(j) = r_\gamma(j) * (\gamma - 1)$. For client j with $F_j^* > 0$ we have $r'_\gamma(j) = \frac{C_j^* - \sum_{i \in \mathcal{F}} c_{ij} \overline{x}_{ij}}{F_j^*}$ which is the fractional connection cost minus the average distance to a close facility, divided by the fractional facility cost of a client j.

Observe, that for every client j the following hold (see Figure 3):

- his average distance to a close facility equals $D_{av}^C(j) = C_j^* - r'_\gamma(j) \cdot F_j^*$,
- his average distance to a distant facility equals $D_{av}^D(j) = C_j^* + r_\gamma(j) \cdot F_j^*$,

– his maximal distance to a close facility is at most the average distance to a distant facility, $D^C_{max}(j) \leq D^D_{av}(j) = C^*_j + r_\gamma(j) \cdot F^*_j$.

Consider the bipartite graph G obtained from the solution $(\overline{x}, \overline{y})$, where each client is directly connected to his close facilities. We will greedily cluster this graph in each round choosing the cluster center to be an unclustered client j with the minimal value of $D^C_{av}(j) + D^C_{max}(j)$. In this clustering, each cluster center has a minimal value of $D^C_{av}(j) + D^C_{max}(j)$ among clients in his cluster.

4 Our new algorithm

Consider the following algorithm $A1(\gamma)$:

1. Solve the LP relaxation of the problem to obtain a solution (x^*, y^*).
2. Scale up the value of the facility opening variables y by a constant $\gamma > 1$, then change the value of the x-variables so as to use the closest possible fractionally open facilities (see Section 3.2).
3. If necessary, split facilities to obtain a complete solution $(\overline{x}, \overline{y})$.
4. Compute a greedy clustering for the solution $(\overline{x}, \overline{y})$, choosing as cluster centers unclustered clients minimizing $D^C_{av}(j) + D^C_{max}(j)$.
5. For every cluster center j, open one of his close facilities randomly with probabilities $\overline{x}_{ij}$.
6. For each facility i that is not a close facility of any cluster center, open it independently with probability $\overline{y}_i$.
7. Connect each client to an open facility that is closest to him.

In the analysis of this algorithm we will use the following result:

Lemma 2. *Given n independent events $e_1, e_2, \ldots, e_n$ that occur with probabilities $p_1, p_2, \ldots, p_n$ respectively, the event $e_1 \cup e_2 \cup \ldots \cup e_n$ (i.e. at least one of e_i) occurs with probability at least $1 - \frac{1}{e^{\sum_{i=1}^n p_i}}$, where e denotes the base of the natural logarithm.*

Theorem 1. *Algorithm $A1(\gamma = 1.67736)$ produces a solution with expected cost $E[cost(SOL)] \leq 1.67736 \cdot F^* + 1.37374 \cdot C^*$.*

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 an open facility within a certain distance with a certain probability. If j is a cluster center, one of his close facilities is open and the expected distance to this open facility is $D^C_{av}(j) = C^*_j - r'_\gamma(j) \cdot F^*_j$.

If j is not a cluster center, he first considers his close facilities (see Figure 4). If any of them is open, the expected distance to the closest open facility is at most $D^C_{av}(j)$. From Lemma 2, with probability $p_c \geq (1 - \frac{1}{e})$, at least one close facility is open.

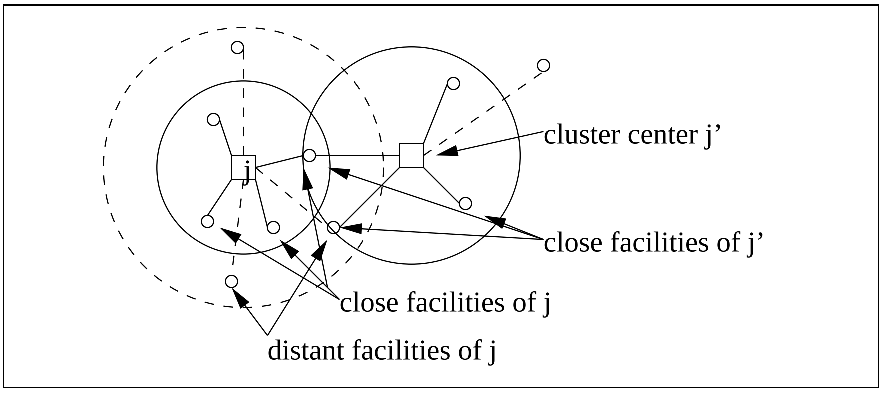

Fig. 4. Facilities that client j may consider: his close facilities, distant facilities, and close facilities of cluster center j'.

Suppose none of the close facilities of j is open, but at least one of his distant facilities is open. Let p_d denote the probability of this event. 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 he connects himself to the facility serving his cluster center $g(j) = j'$. Again from Lemma 2, such an event happens with probability $p_s \leq \frac{1}{e^\gamma}$. In the following we will show that if $\gamma < 2$ then the expected distance from j to the facility serving j' is at most $D_{av}^D(j) + D_{max}^C(j') + D_{av}^C(j')$. Let $\mathcal{C}_j$ $(\mathcal{D}_j)$ be the set of close (distant) facilities of j. For any set of facilities $X \subset \mathcal{F}$, let $d(j, X)$ denote the weighted average distance from j to $i \in X$ (with values of opening variables y_i as weights).

If the distance between j and j' is at most $D_{av}^D(j) + D_{av}^C(j')$, then the remaining $D_{max}^C(j')$ is enough for the distance from j' to any of his close facilities. Suppose now that the distance between j and j' is bigger than $D_{av}^D(j) + D_{av}^C(j')$ (*). We will bound $d(j', \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$, the average distance from cluster center j' to his close facilities that are neither close nor distant facilities of j (since the expected connection cost that we compute is on the condition that j was not served directly). The assumption(*) implies that $d(j', \mathcal{C}_j \cap \mathcal{C}_{j'}) > D_{av}^C(j')$. Therefore, if $d(j', \mathcal{D}_j \cap \mathcal{C}_{j'}) \geq D_{av}^C(j')$, then $d(j', \mathcal{D}_j \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \leq D_{av}^C(j')$ and the total distance from j is small enough.

The remaining case is that $d(j', \mathcal{D}_j \cap \mathcal{C}_{j'}) = D_{av}^C(j') - z$ for some positive z (**). Let $\hat{y} = \sum_{i \in (\mathcal{C}_{j'} \cap \mathcal{D}_j)} \overline{y}_i$ be the total fractional opening of facilities in $\mathcal{C}_{j'} \cup \mathcal{D}_j$ in the modified fractional solution $(\overline{x}, \overline{y})$. From (*) we conclude, that $d(j, \mathcal{D}_j \cap \mathcal{C}_{j'}) \geq D_{av}^D(j) + z$, which implies $d(j, \mathcal{D}_j \setminus \mathcal{C}_{j'}) \leq D_{av}^D(j) - z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}$ (note that (**) implies $(\mathcal{D}_j \setminus \mathcal{C}_{j'}) \neq \emptyset$ and $\gamma - 1 - \hat{y} > 0$), hence
$D_{max}^C(j) \leq D_{av}^D(j) - z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}$. Combining this with assumption (*) we conclude that the minimal distance from j' to a facility in $\mathcal{C}_j \cap \mathcal{C}_{j'}$ is at least
$D_{av}^D(j) + D_{av}^C(j') - D_{max}^C(j) \geq D_{av}^C(j') + z \cdot \frac{\hat{y}}{\gamma - 1 - \hat{y}}$.

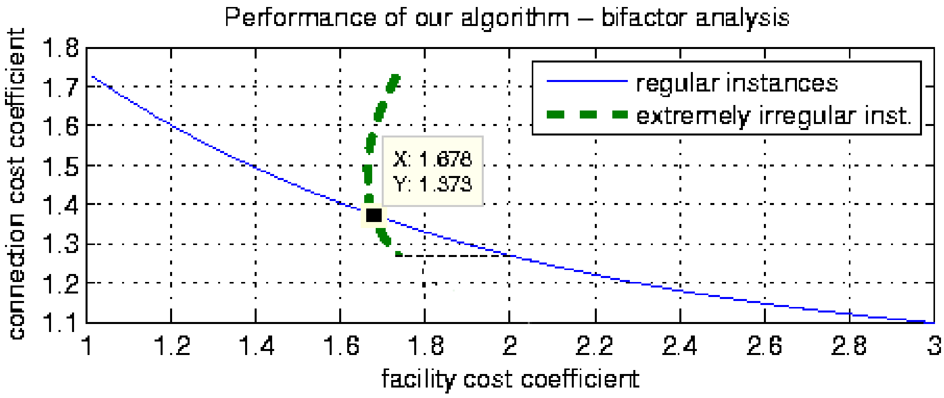

Fig. 5. Figure presents performance of our algorithm for different values of parameter γ. The solid line corresponds to regular instances with $r_\gamma(j) = 0$ for all j and it coincides with the approximability lower bound curve. The dashed line corresponds to instances with $r_\gamma(j) = 1$ for all 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.

Assumption (**) implies $d(j', \mathcal{C}_{j'} \setminus \mathcal{D}_j) = D_{av}^C(j') + z \cdot \frac{\hat{y}}{1-\hat{y}}$.
Concluding, if $\gamma < 2$ then $d(j', \mathcal{C}_{j'} \setminus (\mathcal{D}_j \cup \mathcal{C}_j)) \leq D_{av}^C(j') + z \cdot \frac{\hat{y}}{\gamma-1-\hat{y}}$.
Therefore, the expected connection cost from j to a facility in $\mathcal{C}_{j'} \setminus (\mathcal{D}_j \cup \mathcal{C}_j)$ is at most

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

Putting all the cases together, the expected total connection cost is

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

By setting $\gamma = \gamma_0 \approx 1.67736$ such that $\frac{1}{e} + \frac{1}{e^{\gamma_0}} - (\gamma_0 - 1) \cdot (1 - \frac{1}{e} + \frac{1}{e^{\gamma_0}}) = 0$, we obtain $E[C_{SOL}] \leq (1 + \frac{2}{e^{\gamma_0}}) \cdot C^* \leq 1.37374 \cdot C^*$. $\square$

The algorithm A1 with $\gamma = 1 + \epsilon$ (for a sufficiently small positive ϵ) is essentially the algorithm of Chudak and Shmoys.

5 The 1.5-approximation algorithm

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

In 2002 Jain, Mahdian and Saberi [10] proposed a primal-dual approximation algorithm (the JMS algorithm). Using a dual fitting approach they have shown that it is a 1.61-approximation algorithm. In a later work of Mahdian, Ye and Zhang [12] the following was proven.

Lemma 3 ([12]). *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 2. *Consider the solutions obtained with the A1 and JMS algorithms. The cheaper of them is expected to have a cost at most* 1.5 *times the cost of the optimal fractional solution.*

Proof. Consider the algorithm A2 that with probability $p = 0.313$ runs the JMS algorithm and with probability $1-p$ runs the A1 algorithm. Suppose that you are given an instance, and F^* and C^* are facility and connection costs in an optimal solution to the linear relaxation of the problem for this instance. Consider the expected cost of the solution produced by algorithm A2 for this instance. $E[cost] \leq p{\cdot}(1.11{\cdot}F^*{+}1.7764{\cdot}C^*){+}(1{-}p){\cdot}(1.67736{\cdot}F^*{+}1.37374{\cdot}C^*) = 1.4998 \cdot F^* + 1.4998 \cdot C^* < 1.5 * (F^* + C^*) \leq 1.5 * OPT$. □

Instead of the JMS algorithm we could take the algorithm of Machdian et al. [12] - 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 a notation introduced in Section 2.3, the MYZ(δ) algorithm is the $S_\delta(JMS)$ algorithm. The MYZ(1.504) algorithm was proven [12] 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 (1.67736,1.37374)-approximation algorithm gives a 1.4991-approximation algorithm, which is even better than just using JMS and A1, but it gets more complicated and the additional improvement is tiny.

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, k-th 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 [2] 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 (1.67736,1.37374)-approximation algorithm instead of the JMS algorithm within this framework improves approximation for each level. In particular, in the limit as k tends to ∞ we get 3.236-approximation which is the best possible for this construction.

By a slightly different method, Zhang [17] obtained a 1.77-approximation algorithm for the 2-level problem. By reducing to a problem with smaller number of levels, he obtained 2.523 [1] and 2.81 approximation algorithms for the 3-level and the 4-level version of the problem. If we modiffy the algorithm by Zhang for the 3-level problem, and use the new (1.67736,1.37374)-approximation algorithm for the single level part, we 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].

7 Concluding remarks

The presented algorithm 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 guaranty in 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 guaranty was, however, not necessary to construct the 1.5-approximation algorithm for the metric UFL problem.

With the 1.52-approximation algorithm of Mahdian et al. it was not clear for the authors if a better analysis of the algorithm could close the gap with the approximation lower bound of 1.463 by Guha and Khuler. Byrka and Aardal [3] 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 to move the bifactor approximation guaranty of an algorithm along the approximability lower bound of Jain et al. (see Figure 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.

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

8 Acknowledgments

The author would like to thank Karen Aardal for all her support and many helpful comments on earlier drafts of this paper. The author also thanks David Shmoys, Steven Kelk, Evangelos Markakis and anonymous referees 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 *Information Processing Letters* **72**, pages 161–167, 1999.
2. A. Ageev, Y. Ye, and J. Zhang. Improved combinatorial Approximation algorithms for the k-level facility location problem. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming(ICALP)*, LNCS 2719, pages 145–156, 2003.
3. J. Byrka and K. Aardal. The approximation gap for the metric facility location problem is not yet closed. *Operations Research Letters (ORL)* **35(3)**, pages 379–384, 2007.
4. M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. In *Proc. of the 40th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 378–388, 1999.
5. F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In *Proc. of the 6th Integer Programing and Combinatorial Optimization (IPCO)*, pages 180–194, 1998.
6. F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, **33(1)**, pages 1–25, 2003.
7. 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 Science* **8**, pages 789–810, 1977.
8. S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Proc. of the 9th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 228–248, 1998.
9. D.S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming* **22**, pages 148–162, 1982.
10. K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proc. of the 34th ACM Symp. on Theory of Computing (STOC)*, pages 731–740, 2002.
11. J.-H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation. In *Proc. of the 24th ACM Symp. on Theory of Computing (STOC)*, pages 771–782, 1992.
12. M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Proc. of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 229–242, 2002.
13. D.B. Shmoys, *Approximation algorithms for facility location problems*, in Proc. of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), LNCS 1913, K. Jansen, S. Khuller, eds., Springer, Berlin, 2000, pp. 265–274.

14. D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proc. of the 29th ACM Symp. on Theory of Computing (STOC)*, pages 265–274, 1997.
15. M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *Proc. of the 9th Integer Programming and Combinatorial Optimization (IPCO)*, pages 240–257, 2002.
16. J. Vygen, *Approximation algorithms for facility location problems (Lecture Notes)*, Report No. 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005. `http://www.or.uni-bonn.de/~vygen/fl.pdf`.
17. J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach *Mathematical Programming, Ser. A*, **108**, pages 159-176, 2006.