

An incremental algorithm for the uncapacitated facility location problem

Ashwin Arulsevan^{*1}, Olaf Maurer^{†2}, and Martin Skutella^{‡2}

¹Department of Management Science, University of Strathclyde[§]

²Institut für Mathematik, Technische Universität Berlin

November 13, 2014

Abstract

We study the incremental facility location problem, wherein we are given an instance of the uncapacitated facility location problem and seek an incremental sequence of opening facilities and an incremental sequence of serving customers along with their fixed assignments to facilities open in the partial sequence. We say that a sequence has a competitive ratio of k , if the cost of serving the first ℓ customers in the sequence is at most k times the optimal solution for serving any ℓ customers for all possible values of ℓ . We provide an incremental framework that computes a sequence with a competitive ratio of at most 8 and a worst-case instance that provides a lower bound of 3 for any incremental sequence. We also present the results of our computational experiments carried out on a set of benchmark instances for the uncapacitated facility location problem. The problem has applications in multi-stage network planning.

Keywords: incremental algorithm, facility location problem, network design, competitive ratio, robust location

^{*}ashwin.arulsevan@strath.ac.uk

[†]maurer@math.tu-berlin.de

[‡]martin.skutella@tu-berlin.de

[§]This research was carried out while the author was working at Technische Universität Berlin

1 Introduction

In the uncapacitated facility location problem (UFL), we are given a set of m facilities and a set of n customers. Each facility has an opening cost and each customer-facility pair has an assignment cost. In the robust facility location problem as defined in [2], we are given an additional input $\ell \leq n$ and we seek a cost-minimal solution that connects any ℓ of the n customers to open facilities. The problem was defined by them as robust [2] in the sense that if the instance is altered by adding new customers that are very far away from every facility, the optimal solution would remain the same for the new instance. The presence of a very small fraction of customers who are far away has a significant influence on the service provided to the majority of customers. A final solution might open facilities at absurd locations in order to be favourable to these outlying customers. A robust solution in this sense is impervious to such an effect. We denote the robust facility location problem for a given number ℓ by ℓ -RFLP.

In this work, we study the incremental facility location problem (IncFLP) that is based on the RFLP. The IncFLP asks for sequences of facilities and customers and an assignment of customers to facilities. The customer sequence defines the order in which the customers will be served and the facility sequence defines the order in which the facilities will be opened. The sequence could be viewed as an event happening in time (a facility opened or a customer served). The assignment of a customer to a facility is only valid if the facility was opened earlier in time. Reassignment of customers at a later time point is not permitted and so the solution is incremental.

For every point ℓ in the sequence, we compare the cost of the solution incurred by the sequence to the optimal cost of ℓ -RFLP. We will make these definitions more precise in the next section.

Motivation

The incremental facility location problem is motivated by the need to deploy telecommunication networks in phases due to budget, time or resource restrictions. In practical network planning, problems require incremental solutions that help build the network in stages in the most efficient way. In [2], the authors motivated the RFLP by the presence of a minority of customers that are at a distance from the majority. Connecting the minority requires building facilities exclusively to serve their demands which is not cost effective, especially at the beginning of a network rollout. In practice, a preferential service cannot be offered as a solution since all customers must

be served at some point of time. We also do not have the luxury to build the entire network at a stretch, when a complete solution to the uncapacitated facility location problem is given. By using an incremental solution to this problem, we can identify a subset of the planned network that will be constructed in the early time stages and we can use the resulting profits then to connect more distant parts of the network at a later point of time. This gives us the motivation for our problem. Please observe that this is not the same as building the network for a known preset sequence of customers. We are actually seeking a cost effective sequence. A planner or network builder could choose a specific point in the sequence depending on the resource availability and choose to serve that many customers in the sequence at that point. Subsequent service could be provided for the remaining customers in the sequence at later time periods. The cost of serving the first ℓ customers in the sequence is compared with the cost of optimally serving any ℓ customers, in order to make sure the sequence is cost effective.

Previous work

The uncapacitated facility location problem (UFLP) with metric assignment cost has a long line of research [1, 5, 6, 16]. The best approximation guarantee known for this problem is 1.488 [12] and there can be no polynomial time algorithm to approximate it within a ratio of 1.463 unless $NP \subset DTIME(n^{O(\log \log n)})$ [9]. The problem has also been studied in incremental settings [13, 15] but prior work involves different settings than the one considered in our work.

The author of [15] is interested in a nested sequence of facilities and a threshold sequence for the scaling of the assignment costs. A solution in the sequence is a solution corresponding to a specific scaling factor, which could be inferred from the threshold sequence. Each solution in the sequence serves all customers. The nested sequence of facilities gives a near optimal solution for every possible scaling factor. Lin et al. [13] provided a framework for solving several incremental problems and solved the problem studied in [15] under the same settings. The framework proposed by them uses a black box algorithm to iteratively augment partial competitive solutions to reach subsequent competitive solutions that eventually become complete. Chobrak et al. [3, 4] independently obtained competitive deterministic and randomised algorithms for the incremental k -median problem. The results are obtained through a reduction to the online bidding problem. We follow the doubling technique proposed in the above works. This involves incrementally building a solution by comparing it with the optimal solution

to ensure competitiveness and refining it when it is necessary. Since we are required to produce a sequence of customers and facilities in our solution, we deviate slightly by saving the partially refined solutions and delay the construction of incremental solutions. We run a second phase to construct the incremental solution from the partially saved refined solutions.

In [7], the authors are interested in an incremental solution when the demand points arrive one at a time assuming we have complete knowledge of the network. A constant factor competitive algorithm is proposed assuming uniform facility cost. In this algorithm, merging of existing clusters is allowed. This is in contrast to an online algorithm in [14], where the decision to be served by a facility is irrevocable. An $O(\log n)$ -competitive algorithm was proposed for this version of the online problem. This was later improved to $O(\frac{\log n}{\log \log n})$ in [8]. In both works, the algorithm is compared against an offline algorithm that is aware of the arrival order of the customers and the competitive ratio is calculated for all possible arrival orders.

The robust version of the facility location problem (RFLP) was first studied by Charikar et al. [2]. They gave a 3-approximation algorithm using the primal-dual technique. This approximation guarantee was later improved to 2 by Jain et al. [11]. We will be using this as a black box algorithm in our framework.

Contribution

We provide an algorithm that produces a sequence that is within a factor of 8 from the optimal ℓ -RFLP for $\ell = 1, \dots, |R|$. We then present the computational results of our implementation of the framework on a set of benchmark instances for UFL. For each instance, we give the worst-case and average-case bounds of our incremental sequence (for $\ell = 1, \dots, n$) constructed by the algorithm when compared against the optimal cost of ℓ -RFLP.

Organization

We begin in Section 2 with some basic notation and the formal problem definition. In Section 3, we provide the algorithmic framework and explain it. In Section 4, we give the analysis of the algorithm and show it is 8-competitive. In Section 5, we present an example that provides a lower bound on the best possible factor that could be achieved by any algorithm. We present our computational results in Section 6. Finally we conclude with a summary and notes on future extensions in Section 7.

2 Problem definition

We are given a set F of potential facility locations and a set R of customers. Also, we are given metric service costs $c_S : F \times R \rightarrow \mathbb{Q}_+$ and facility opening costs $c_F : F \rightarrow \mathbb{Q}_+$. Let $m = |F|$ be the number of potential facilities and $n = |R|$ the number of customers. We seek nested sets of facilities, customers and assignment edges between them for $\ell = 1, \dots, n$. An approximation algorithm to this problem is called a k -competitive algorithm if the costs of the induced solutions of the ℓ -RFLPs are not more than a factor of k away from the costs of optimal solutions to the ℓ -RFLPs, where k does not depend on ℓ and not on the instance.

We need additional notation to make this precise. For a subset of facilities $F' \subseteq F$, we denote the total facility cost of this set by $c_F(F')$. We denote the service cost incurred by serving a set of customers $R' \subseteq R$ by a set of facilities $F' \subseteq F$ by $c_S(F', R') = \sum_{j \in R'} c_S(F', j)$, where $c_S(F', j)$ is the cost of the cheapest assignment of customer j to a facility in F' . For a number r less than $|R'|$, let $c_S(F', R', r)$ denote the cost of the cheapest assignment of r customers from R' to a facility in F' .

In the IncFLP, we seek a sequence to serve customers

$$\emptyset = R_0 \subsetneq R_1 \subsetneq R_2 \subsetneq \dots \subsetneq R_n = R$$

where $|R_\ell| = \ell$ and a sequence to open facilities

$$F_1 \subseteq F_2 \subseteq \dots \subseteq F_n$$

such that the customers in $R_\ell \setminus R_{\ell-1}$ can be assigned to facilities in F_ℓ . The objective is to minimize the competitive ratio of the sequence. The competitive ratio of a sequence is defined as

$$\max_{\ell=1, \dots, n} \frac{c_F(F_\ell) + \sum_{k=1}^{\ell} c_S(F_k, R_k \setminus R_{k-1})}{OPT_\ell},$$

where OPT_ℓ is the optimal cost of the ℓ -RFLP.

3 Algorithm $FacInc(F, R)$

Let A be any approximation algorithm for the RFLP. We will use A as a black-box. We write $(Z, M) = A(F, R, \ell)$ and mean that A takes as input the set of potential facility locations F , the set of customers R and an integer $\ell \leq n$, where ℓ is the number of customers to be served. It produces the

output (Z, M) , where $Z \subseteq F$ is the set of facilities opened and $M \subseteq R$, $|M| = \ell$, is the set of customers served by the facilities in Z . Given a set of customers and the set of open facilities, an optimal assignment could be easily computed by assigning each customer in the given set to its closest open facility. We now provide our incremental framework $FacInc(F, R)$.

We start with a solution to the UFLP obtained from A with $\ell := n$. Let $F' \subseteq F$ be the set of facilities opened in this solution. Our algorithm runs in two phases, the refinement phase and the incremental phase. The refinement phase constructs and collects solutions to the ℓ -RFLP. These solutions are not necessarily incremental. In the incremental phase, these partial solutions are glued together to construct an incremental solution. The point at which we collect a new partial solution in the refinement phase is called a refinement point. The initial complete solution for the UFLP will be the first refinement point solution we collect. We iteratively reduce the value of ℓ from $|R|$ to 1 and check if the last collected refinement point solution cost exceeds twice the cost of the solution $A(F, R, \ell)$. If it does, then we collect the $A(F, R, \ell)$ as a new refinement point solution.

Refinement phase:

- 1: Initialize: $F_c := F'$, $R_c := R$, $P := \{n\}$
- 2: **for** $\ell = (n - 1)$ to 1 **do**
- 3: $(F_A, R_A) := A(F, R, \ell)$
- 4: **if** $c_F(F_c) + c_S(F_c, R_c) \geq 2(c_F(F_A) + c_S(F_A, R_A))$ **then**
- 5: $F_\ell := F_c := F_A$
- 6: $R_\ell := R_c := R_A$
- 7: $P := P \cup \{\ell\}$
- 8: **end if**
- 9: **end for**

The elements in the set P correspond to the points at which refinement took place. Let their total number $|P| = K$. We shall now index the refinement point solutions from 1 to K in the increasing order of the number of customers they are serving. This helps us in presenting the analysis with clarity. Let us denote the solutions as $(F_1, R_1), (F_2, R_2), \dots, (F_K, R_K)$; notice that $F_K = F'$ and $R_K = R$.

For $k = 1, \dots, K$, let $r_k := |R_k|$ be the number of customers served in the k^{th} refinement point solution. We call the time between the service commencement of customer $(r_{k-1} + 1)$ and r_k as the k^{th} period (with $r_0 := 0$). We also say this period belongs to the solution (F_k, R_k) . From a solution (F_k, R_k) at a refinement point, we can construct solutions with a constant approximation ratio for its period.

In the incremental phase, we glue these partial refined solutions together to construct an incremental solution. We will do this by opening all facilities belonging to a period at the period's beginning.

Incremental phase:

- 1: Initialize: $\tilde{F}_1 := F_1, k := 2$
- 2: **for** $\ell = 2$ to n **do**
- 3: **if** $(\ell - 1) \in P$ **then**
- 4: $\tilde{F}_\ell := \tilde{F}_{\ell-1} \cup F_k$
- 5: $k := k + 1$
- 6: **else**
- 7: $\tilde{F}_\ell := \tilde{F}_{\ell-1}$
- 8: **end if**
- 9: **end for**

Notice that $\tilde{F}_{r_k} = \bigcup_{j=1}^k F_j$ and we now have our incremental sequence of facilities:

$$\tilde{F}_1 \subseteq \tilde{F}_2 \subseteq \dots \subseteq \tilde{F}_{n-1} \subseteq \tilde{F}_n = \bigcup_{k=1}^K F_k .$$

As for customers, we can pick an incremental sequence greedily by adding in every step the current non-served customer with cheapest service cost with respect to the currently open facilities. This yields a sequence

$$\emptyset = \tilde{R}_0 \subsetneq \tilde{R}_1 \subsetneq \dots \subsetneq \tilde{R}_{n-1} \subsetneq \tilde{R}_n = R .$$

4 Analysis

We first show that, at refinement points, the cost of our incremental solution is at most twice the cost of the solution provided by Algorithm A.

Theorem 4.1. *For all refinement points $k = 1, \dots, K$, the cost of our incremental solution for r_k customers is at most $2c_F(F_k) + 2c_S(F_k, R_k)$.*

To this end, we first show the following lemma, which bounds the increase in cost from one refinement point to the next.

Lemma 4.2. *For all refinement points $k = 2, \dots, K$,*

$$\begin{aligned} 2c_F(F_{k-1}) + 2c_S(F_{k-1}, R_{k-1}) + c_F(F_k) + c_S(F_k, R_k \setminus \tilde{R}_{r_{k-1}}, r_k - r_{k-1}) \\ \leq 2c_F(F_k) + 2c_S(F_k, R_k) . \end{aligned}$$

Proof. Our refinement condition at step 4 of the incremental phase implies

$$2c_F(F_{k-1}) + 2c_S(F_{k-1}, R_{k-1}) < c_F(F_k) + c_S(F_k, R_k) .$$

This yields

$$\begin{aligned} 2c_F(F_{k-1}) + 2c_S(F_{k-1}, R_{k-1}) + c_F(F_k) + c_S(F_k, R_k \setminus \tilde{R}_{r_{k-1}}, r_k - r_{k-1}) \\ < 2c_F(F_k) + c_S(F_k, R_k \setminus \tilde{R}_{r_{k-1}}, r_k - r_{k-1}) + c_S(F_k, R_k) \\ \leq 2c_F(F_k) + 2c_S(F_k, R_k) . \end{aligned}$$

This completes the proof. \square

We are now ready to bound the additional cost incurred by the solution being incremental.

Lemma 4.3. *For all refinement points $k = 1, \dots, K$,*

$$\begin{aligned} 2c_F(F_1) + 2c_S(F_1, R_1) + \sum_{j=2}^k \left(c_F(F_j) + c_S(F_j, R_j \setminus \tilde{R}_{r_{j-1}}, r_j - r_{j-1}) \right) \\ \leq 2c_F(F_k) + 2c_S(F_k, R_k) . \end{aligned}$$

Proof. For $k = 1$, the statement is trivial. For $k = 2$, the statement follows directly from Lemma 4.2. We proceed by induction on k . Assume the statement is true for $k - 1$. Then we get

$$\begin{aligned} 2c_F(F_1) + 2c_S(F_1, R_1) + \sum_{j=2}^k \left(c_F(F_j) + c_S(F_j, R_j \setminus \tilde{R}_{r_{j-1}}, r_j - r_{j-1}) \right) \\ = 2c_F(F_1) + 2c_S(F_1, R_1) + \sum_{j=2}^{k-1} \left(c_F(F_j) + c_S(F_j, R_j \setminus \tilde{R}_{r_{j-1}}, r_j - r_{j-1}) \right) \\ \quad + c_F(F_k) + c_S(F_k, R_k \setminus \tilde{R}_{r_{k-1}}, r_k - r_{k-1}) \\ \leq 2c_F(F_{k-1}) + 2c_S(F_{k-1}, R_{k-1}) + c_F(F_k) + c_S(F_k, R_k \setminus \tilde{R}_{r_{k-1}}, r_k - r_{k-1}) \\ \leq 2c_F(F_k) + 2c_S(F_k, R_k) , \end{aligned}$$

where the last inequality follows from Lemma 4.2 again. \square

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Since $\tilde{F}_{r_k} = \bigcup_{j=1}^k F_j$ we can bound the facility opening cost by

$$c_F(\tilde{F}_{r_k}) \leq \sum_{j=1}^k c_F(F_j) \leq 2c_F(F_1) + \sum_{j=2}^k c_F(F_j) .$$

Moreover, by the greedy construction of the sequence of customers, we can bound the connection cost by

$$c_S(F_1, R_1) + \sum_{j=2}^k c_S(F_j, R_j \setminus \tilde{R}_{r_{j-1}}, r_j - r_{j-1}) .$$

The desired bound thus follows from Lemma 4.3. \square

It remains to analyze the performance of our incremental solution for an arbitrary number of customers ℓ , where ℓ is not necessarily a refinement point. The next theorem is an easy consequence of Theorem 4.1 and the refinement condition at step 4 of the incremental phase.

Theorem 4.4. *For $\ell = 1, \dots, n$, the cost of the incremental solution for ℓ customers is at most $4c(A(F, R, \ell))$, where $c(A(F, R, \ell))$ denotes the cost of the solution found by Algorithm A.*

Theorem 4.4 implies that our algorithm yields an 8-competitive incremental solution if we plug in the best known 2-approximation algorithm of Jain et al. [11] for the RFLP. If we are willing to spend more time and solve the RFLP exactly, we obtain a 4-competitive incremental solution with our framework.

5 Lower bound

We describe an instance for which no incremental solution can be better than 3-competitive. More precisely, we give a family of instances with more than 200 facilities which yield a lower bound of at least 2.99. The ratio for this construction does not seem to exceed a value of 3.

Let there be m facilities. Each facility $i = 1, \dots, m$ has zero-cost-connections to a set R_i of 2^{i-1} customers. The sets R_i are mutually disjoint. Additionally, every facility j is connected to every customer in the set R_i , for all $i \neq j$, with a very high assignment cost $M \gg 0$; note that the connection costs would still induce a metric. We refer to the facility together with its customers of zero service cost as a cluster.

Let x_1, \dots, x_m be the facility opening costs. Consider the following system of linear inequalities:

$$x_{i+1} + \sum_{j=1}^{i-1} x_j \geq \alpha \cdot x_i \quad \text{for } i = 1, \dots, m-1.$$

Here α is the achieved minimal competitive ratio by these inequalities. As an example, for $m = 4$ this system is feasible for $\alpha = 2.246$, but infeasible for $\alpha = 2.247$. So by setting the facility costs to a solution of this system, we get a lower bound of 2.246.

The intuitive explanation behind these inequalities is the following. Suppose one looks at a point in any sequence that gives a solution serving 2^{i-1} customers. To keep the competitive ratio of the sequence below α for serving just one customer, one needs to first open facility 1 and serve the single customer in R_1 , as cluster 2 is α times as expensive. By proceeding with such an inductive reasoning, the above inequalities give an incentive to open the clusters one by one in the order of their size, as the competitive ratio for the sequence is immediately as bad as α as soon as one skips a cluster. On the other hand, the optimal solution to serve 2^{i-1} customers is to open facility i and serve customers in the set R_i . The value α should now be chosen carefully, so that the ratio between the solution provided by the sequence in opening all facilities from $1, \dots, i$ to that of opening facility i alone is at least α for some i , i.e., $\alpha \leq \max_{i=1, \dots, m} \frac{\sum_{j=1}^i x_j}{x_i}$. An example for $m = 4, \alpha = 2.246$ can be seen in Figure 1. As we increase the value of m , the value of α increases as well and it numerically converges to 3. For a value of $m = 200$, we get $\alpha \approx 2.99$.

As one could easily note that the RFLP problem can be solved exactly for the lower bound instances provided in this section in polynomial time, our incremental algorithm would achieve a competitive factor of 4 for these instances.

6 Experiments

We tested the quality of the incremental solution produced by our algorithm against the optimal (non-incremental) solution. This is the best possible solution for serving $\ell = 1, \dots, n$ customers. We present the maximum and average gap over all $\ell = 1, \dots, n$ for every instance. We also report the running time of the algorithms. The experiments were carried out on a i7-4771 machine with 16GB of RAM. We used Gurobi Optimizer 5.6 to obtain

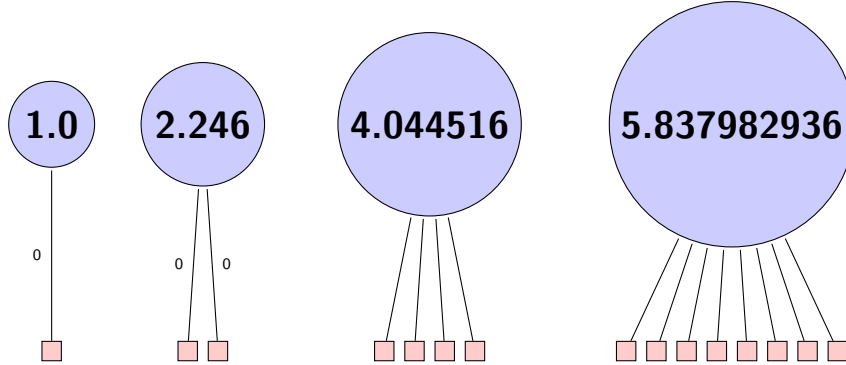


Figure 1: Lower bound example for $m = 4$, $\alpha = 2.246$. All assignment costs are zero.

the optimal solution of ℓ -RFLP. The instances were the set of benchmark instances for the uncapacitated facility location problem from the UfLib library provided by Hofer [10]. All instances are complete bipartite graphs and in all instances, the number of facilities equals the number of customers.

The results from our computational study are presented in Table 1. For each instance, we provide the size of the instance and the worst-case gap for some $\ell = 1, \dots, n$ and the average gap over all customers. Note that we are comparing our solution against the optimal solution of serving any ℓ customers and this is not an incremental solution. As pointed out in Section 5, we need to pay a price for obtaining an incremental solution. In almost all instances, the worst-case gap is about 50% (corresponding to a factor of 1.5 from the optimal) and the average gap is less than 15% for all instances. We also report the running time (in seconds) of our implementation. All instances took less than an hour.

7 Conclusions

In this work, we presented an incremental framework to provide a sequence of facilities to be opened and customers to be served along with their fixed assignments that provides a constant factor competitive guarantee. We prove a lower bound of 3 through an example. We gave an implementation of the algorithm and the competitive factor on a set of benchmark instances was computed and presented. A number of questions are still open. Is the analysis tight for the provided algorithm or can the worst-case guarantee be

Instance	# Fac/Cust	Max gap (%)	Ave Gap (%)	Time[sec]
c10-mp1	200	59.50	15.88	286.25
c10-mp2	200	50.65	13.73	315.34
c10-mq1	300	40.06	13.41	2048.95
c10-mq2	300	40.06	11.38	2057.54
c15-mp1	200	59.50	15.88	284.62
c15-mp2	200	50.65	13.73	316.19
c15-mq1	300	40.06	13.41	2076.61
c15-mq2	300	40.06	11.38	2077.86
c20-mp1	200	59.50	15.88	289.52
c20-mp2	200	50.65	13.73	319.90
c20-mq1	300	40.06	13.41	2052.22
c20-mq2	300	40.06	11.38	2061.86
c5-mp1	200	59.50	15.88	287.77
c5-mp2	200	50.65	13.73	316.01
c5-mq1	300	40.06	13.41	2058.01
c5-mq2	300	40.06	11.38	2060.26
d10-mp1	200	59.50	15.88	285.44
d10-mp2	200	50.65	13.73	316.20
d10-mq1	300	40.06	13.41	2059.18
d10-mq2	300	40.06	11.38	2071.02
d15-mp1	200	59.50	15.88	285.26
d15-mp2	200	50.65	13.73	315.42
d15-mq1	300	40.06	13.41	2049.25
d15-mq2	300	40.06	11.38	2058.37
d20-mp1	200	59.50	15.88	287.47
d20-mp2	200	50.65	13.73	317.52
d20-mq1	300	40.06	13.41	2216.26
d20-mq2	300	40.06	11.38	2080.21
d5-mp1	200	59.50	15.88	286.38
d5-mp2	200	50.65	13.73	316.51
d5-mq1	300	40.06	13.41	2054.96
d5-mq2	300	40.06	11.31	2079.71

Table 1: Results of computational experiments

improved with this or other algorithms? Can the lower bound be increased? Another interesting direction of future research is to study our incremental setting for other variations of the facility location problem that arise in practice, such as the buy-at-bulk network design problems and connected facility location problems.

Acknowledgements

We thank the anonymous reviewers in helping us improve our presentation. This research work was supported by the DFG Research Center MATHEON “Mathematics for key technologies” in Berlin.

References

- [1] J. Byrka and K. Aardal, An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem, *SIAM J. Comput* 39 (2010), 2212–2231.
- [2] M. Charikar, S. Khuller, D.M. Mount, and G. Narasimhan, Algorithms for facility location problems with outliers, *Proc 12th Ann ACM-SIAM Symp Discr Algorithms*, Washington, DC, 2001, pp. 642–651.
- [3] M. Chrobak, C. Kenyon, J. Noga, and N. Young, Incremental medians via online bidding, *Algorithmica* 50 (2008), 455–478.
- [4] M. Chrobak and C. Kenyon-Mathieu, SIGACT news online algorithms column 10: Competitiveness via doubling, *SIGACT News* 37 (2006), 115–126.
- [5] F.A. Chudak and D.B. Shmoys, Improved approximation algorithms for the uncapacitated facility location problem, *SIAM J. Comput* 33 (2003), 1–25.
- [6] G.P. Cornuéjols, G.L. Nemhauser, and L.A. Wolsey, “The uncapacitated facility location problem,” *Discrete location theory*, P. Mirchandani and R. Francis (Editors), John Wiley and Sons, Inc., New York, 1990, pp. 119–171.
- [7] D. Fotakis, Incremental algorithms for facility location and k -median, *Theor Comput Sci* 361 (2006), 275–313.

- [8] D. Fotakis, On the competitive ratio for online facility location, *Algorithmica* 50 (2008), 1–57.
- [9] S. Guha and S. Khuller, Greedy strikes back: Improved facility location algorithms, *Proc 9th Ann ACM-SIAM Symp Discr Algorithms*, San Francisco, CA, 1998, pp. 649–657.
- [10] M. Hoefer, UflLib, webpage. <http://resources.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/>.
- [11] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani, Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP, *J. ACM* 50 (2003), 795–824.
- [12] S. Li, A 1.488-approximation algorithm for the uncapacitated facility location problem, *Proc 38th Ann Int Colloq Automata, Languages Program, Lecture Notes in Computer Science*, Vol. 6756, Springer, 2011, pp. 77–88.
- [13] G. Lin, C. Nagarajan, R. Rajaraman, and D.P. Williamson, A general approach for incremental approximation and hierarchical clustering, *SIAM J. Comput* 39 (2010), 3633–3669.
- [14] A. Meyerson, Online facility location, *Proc 42nd Ann IEEE symposium Symp Foundations Comput Sci*, Las Vegas, NV, 2001, pp. 426–431.
- [15] C.G. Plaxton, Approximation algorithms for hierarchical location problems, *Proc 35th Ann ACM Symp Theory Comput*, San Diego, CA, 2003, pp. 40–49.
- [16] D. Shmoys, É. Tardos, and K. Aardal, Approximation algorithms for facility location problems, *Proc 29th Ann ACM Symp Theory Comput*, El Paso, TX, 1997, pp. 265–274.