Online Non-metric Facility Location with Service Installation Costs

Christine Markarian
Department of Engineering and Information Technology, University of Dubai, Dubai, U.A.E.

Keywords: Facility Location Problem, Service-installation Costs, Online Algorithms, Competitive Analysis, Randomized Rounding.

Abstract: In this paper, we study the non-metric *Online Facility Location with Service Installation Costs* problem (OFL-SIC), an extension of the well-known non-metric *Online Facility Location* problem. In OFL-SIC, we are given a set of facilities, a set of services, and a set of requests arriving over time. Each request is composed of a subset of the services. Facilities are enabled to offer a subset of the services when being opened and an algorithm has to ensure that each arriving request is connected to a set of open facilities jointly offering the requested services. Opening a facility incurs an *opening cost* and for each offered service, there is a *service installation cost* that needs to be paid if the algorithm decides to install the service at the facility. Connecting a request to an open facility incurs a *connecting cost*, which is equal to the distance between the request and the facility. The goal is to minimize the total opening, service installation, and connecting costs. We propose the first online algorithm for non-metric OFL-SIC and show that it is asymptotically optimal under the standard notion of *competitive analysis* which is used to evaluate the performance of online algorithms.

1 INTRODUCTION

With the rapid growth in urbanization, the demand for different services in cities has been significantly increased. It is estimated that by the year 2050, about 66 percent of the world's population will be living in an urban environment (World Urbanization Prospects, 2014). With this trend, cities are expected to build facilities that offer various services not only to the people currently residing in the city but also to the people who are likely to move there in the future. Without adequate information about the future, in regards to how many people will move, when, and what services they will request, decisions about where to locate facilities and what services to assign to each facility, with minimum possible costs, become more challenging. At the heart of such decisions lie complex facility location optimization problems, which we approach in this paper from an algorithmic perspective. In particular, we are interested in developing algorithms that can be *proven* to be optimal or best possible.

We consider the *Facility Location with Service Installation Costs* problem (FL-SIC), which is an extension of the well-studied Facility Location problem (FL). FL-SIC was introduced by Shmoys *et al.* (Shmoys et al., 2004) and is defined as follows. We are given a set of facilities, a set of services, and a set of requests. Each request is composed of a subset of services. Facilities are enabled to offer a subset of the services when being opened and an algorithm has to ensure that a request is connected to a set of open facilities jointly offering the requested services. Opening a facility incurs an *opening cost* and for each offered service, there is a *service installation cost* that needs to be paid if the algorithm decides to install the service at the facility. Connecting a request to an open facility incurs a *connecting cost*, which is equal to the distance between the request and the facility. The goal is to minimize the total opening, service installation, and connecting costs.

Shmoys *et al.* (Shmoys et al., 2004) studied the *metric* version of FL-SIC in which facilities and requests are assumed to reside in a metric space. A more general variant of FL-SIC, known as the *Multi-Commodity Facility Location* problem (MCFL), has been studied for both the metric (Poplawski and Rajaraman, 2011; Ravi and Sinha, 2010; Shmoys et al., 2004; Svitkina and Tardos, 2010) and non-metric (Fleischer et al., 2006) versions. In MCFL, one facility cost is given for opening a facility *and* installing its services. In FL-SIC, facility costs are split, so as we pay a fixed cost for opening a facility and then for each service installed at the facility, we pay a service cost associated with it. FL-SIC is also known as *MCFL with linear costs*. FL-SIC becomes a special case of MCFL if we set each facility cost (in MCFL) to the sum of opening and service installation costs (in FL-SIC).

In many real-world scenarios, predicting future events is difficult if not impossible, and yet, we are

Markarian, C.
Online Non-metric Facility Location with Service Installation Costs.
DOI: 10.5220/0010469207370743
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 737-743
ISBN: 978-989-758-509-8

expected to make immediate wise decisions with as few regrets as possible. From an algorithmic perspective, these scenarios are modeled as *online* problems and are solved with *online* algorithms. Motivated by such scenarios, we study the online variant of FL-SIC which we refer to as the non-metric *Online Facility Location with Service Installation Costs* problem (OFL-SIC), defined as follows.

Definition 1. (Non-metric Online Facility Location with Service Installation Costs) *We are given a set of facilities, a set of services, and a set of requests arriving over time. Each request is composed of a subset of the services. Facilities are enabled to offer a subset of the services when being opened and an algorithm has to ensure that each arriving request is connected to a set of open facilities jointly offering the requested services. All decisions are to be made irrevocably. Opening a facility incurs an opening cost and for each offered service, there is a service installation cost that needs to be paid if the algorithm decides to install the service at the facility. Connecting a request to an open facility incurs a connecting cost, which is equal to the distance between the request and the facility. The goal is to minimize the total opening, service installation, and connecting costs.*

To the best of our knowledge, no online algorithm exists for this variant.

2 OUR CONTRIBUTION

We propose the first online algorithm for non-metric OFL-SIC and analyze it under the standard notion of *competitive ratio*. An online algorithm is c-competitive or has competitive ratio c if for all sequences of demands, the cost incurred by the algorithm is at most c times the cost incurred by an optimal offline algorithm, which knows the entire sequence of demands in advance.

Our algorithm is based on *randomized rounding*, a technique commonly used in the design and analysis of online algorithms. It has an $O(\log(nk)\log m)$-competitive ratio, where m is the number of facilities, n is the number of requests, and k is the number of services. The competitive analysis of our algorithm is based on ideas taken from the competitive analysis of Meyerson for the *Parking Permit* problem (Meyerson, 2005).

We also show that the competitive ratio of our algorithm is asymptotically optimal by giving a lower bound of $\Omega(\log(nk)\log m)$ on the competitive ratio of any polynomial-time randomized algorithm for non-metric OFL-SIC, under the assumption that NP $\not\subseteq$ BPP. The latter is the result of two reductions from the *Online Set Cover* problem (OSC), which was introduced by Alon *et al.* (Alon et al., 2009). Korman (Korman, 2005) gave an $\Omega(\log m \log n)$ lower bound on the competitive ratio of any polynomial-time randomized algorithm for OSC, under the assumption that NP $\not\subseteq$ BPP.

Outline. The rest of the paper is structured as follows. In Section 3, we give an overview of related works. In Section 4, we present a lower bound on the competitive ratio of any online polynomial-time randomized algorithm for non-metric OFL-SIC. In Section 5, we describe our online algorithm for non-metric OFL-SIC and give its competitive analysis in Section 6. We conclude with some open problems in Section 7.

3 RELATED WORK

In the offline setting, Fleischer *et al.* (Fleischer et al., 2006) studied the non-metric *Multi-Commodity Facility Location* problem (MCFL) and gave an approximation ratio logarithmic in the number of requests, facility locations, and services. As for metric MCFL, there were many works, mostly varying in the facility cost function (Poplawski and Rajaraman, 2011; Ravi and Sinha, 2010; Shmoys et al., 2004; Svitkina and Tardos, 2010), including that of Shmoys *et al.* (Shmoys et al., 2004) for the *Facility Location with Service Installation Costs* problem (FL-SIC).

In the online setting, only the metric version of the *Online Multi-Commodity Facility Location* problem (OMCFL), the online variant of MCFL, has been studied, by Castenow *et al.* (Castenow et al., 2020). Castenow *et al.* (Castenow et al., 2020) gave deterministic and randomized online algorithms along with a lower bound on the competitive ratio of any randomized online algorithm for metric OMCFL.

A closely related online problem is the non-metric *Online Facility Location* problem (OFL), the online variant of non-metric FL, due to Alon *et al.* (Alon et al., 2006). Non-metric OFL is a special case of non-metric OFL-SIC in which the number of services is 1. Alon *et al.* (Alon et al., 2006) proposed a randomized algorithm for non-metric OFL, with asymptotically optimal $O(\log m \log n)$-competitive ratio, where m is the number of facilities and n is the number of clients. The metric variant of *Online Facility Location* has been intensively studied. Meyerson (Meyerson, 2001) introduced a randomized algorithm with $O(\log n)$-competitive ratio. Fotakis (Fotakis, 2003) later proved that the algorithm is $O(\frac{\log n}{\log\log n})$-competitive and showed that this is the best possible competitive ratio for any online algorithm. He also gave a deterministic algorithm with the same competitive ratio. Fotakis (Fotakis, 2007) also provided a simpler online algorithm with an $O(\log n)$-competitive ratio.

4 LOWER BOUND

In this section, we give an $\Omega(\log(nk)\log m)$ lower bound on the competitive ratio of any online polynomial-time randomized algorithm for non-metric OFL-SIC, under the assumption that NP $\not\subseteq$ BPP.

The lower bound is achieved by two reductions from the *Online Set Cover* problem (OSC), introduced by Alon *et al.* (Alon et al., 2009), and defined as follows.

Definition 2. (Online Set Cover). *We are given a universe of n elements and m subsets of this universe, each associated with a cost. Elements are revealed to the algorithm over time and as soon as one arrives, the online algorithm needs to make sure that there is at least one subset purchased that contains the element. The goal is to minimize the total cost of subsets purchased.*

Korman (Korman, 2005) gave an $\Omega(\log m \log n)$ lower bound on the competitive ratio of any polynomial-time randomized algorithm for OSC, where m is the number of subsets and n is the number of elements, under the assumption that NP $\not\subseteq$ BPP.

Our lower bound is based on the following two observations.

Observation 1. *No randomized online polynomial-time algorithm for the non-metric Online Facility Location with Service Installation Costs problem (OFL-SIC) can achieve a competitive ratio better than* $\Omega(\log m \log n)$*, under the assumption that NP* $\not\subseteq$ *BPP, where m is the number of facilities and n is the number of requests.*

Proof. Let I be an instance of Online Set Cover. We transform I into an instance I' of non-metric OFL-SIC as follows. We set the number of services to 1. We represent each subset of I as a facility offering the service and set its opening cost to the corresponding subset cost. Service installation costs are all set to 0. We represent each element as a request with the (one) service. For each request, we let the distance from the request to the facility be 0 if the corresponding element belongs to the subset, and infinity otherwise. Now, every solution to I' corresponds to a solution to I of the same cost, and vice versa. Due to the lower bound of Korman (Korman, 2005), we can imply the result, where n is the number of requests (elements) and m is the number of facilities (subsets). □

Observation 2. *No randomized polynomial-time online algorithm for the non-metric Online Facility Location with Service Installation Costs problem (OFL-SIC) can achieve a competitive ratio better than* $\Omega(\log m \log k)$*, under the assumption that NP* $\not\subseteq$ *BPP, where m is the number of facilities and k is the number of services.*

Proof. Let I be an instance of Online Set Cover. We transform I into an instance I' of non-metric OFL-SIC as follows. Each element is represented as a service. Each subset is represented as a facility with opening cost equal to the corresponding subset cost and offering the services of the corresponding elements. Service installation costs are all set to 0. In each step, a request with the (one) service arrives. In the I' instance, the algorithm is allowed to open any facility and install at it any subset of the services. The OSC instance I need not include all such combinations and so for each facility with a subset of services that is not yet created, we create it and set its opening cost to infinity. Now, every solution to I' corresponds to a solution to I of the same cost, and vice versa. Due to the lower bound of Korman (Korman, 2005), we can imply the result, where k is the number of services (elements) and m is the number of facilities (subsets). □

By combining the two observations, we conclude the following.

Theorem 1. (Lower Bound). *No online polynomial-time randomized algorithm for the non-metric Online Facility Location with Service Installation Costs problem (OFL-SIC) can achieve a competitive ratio better than* $\Omega(\log(nk)\log m)$*, where m is the number of facilities, n is the number of requests, and k is the number of services, under the assumption that NP* $\not\subseteq$ *BPP.*

5 ONLINE ALGORITHM

In this section, we present an online randomized algorithm for non-metric OFL-SIC.

We formulate non-metric OFL-SIC as a directed edge-weighted graph, as follows. We generate a node for each of the k services, called *service nodes*. For each facility, we generate a node, called *facility nodes*. We add an edge from each facility node to each service node. Each edge from a facility node to a service node has weight equal to the corresponding service installation cost. We make a copy of each facility node, called *duplicate facility nodes*. Each facility node will be connected to its duplicate through an edge of weight equal to the opening cost of the facility.

Since requests are not given all at once, a *request node* will be created as soon as a request arrives. We add an edge from the request node to each facility duplicate node and set its weight to the distance between the request and the facility (the connecting cost).

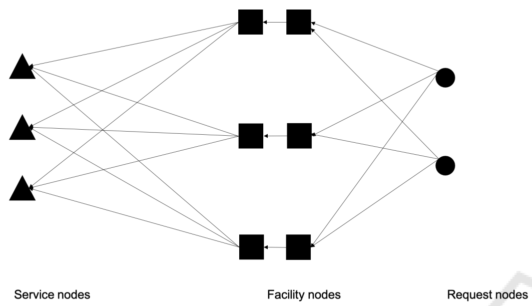

Figure 1: Graph formulation of Non-metric OFL-SIC.

Figure 1 gives an example of two requests, three facilities, and three services. Assume request r arrives and asks for the services 1 and 3. To serve r, the algorithm needs to find two paths: a path from the request node r to the service node 1 and another path from the request node r to the service node 3. It is easy to see that the edges on these paths represent a feasible solution for r. Let p be a solution path. p would contain an edge that corresponds to some duplicate facility - the algorithm will connect the client to that facility and open the facility, if it is not already open. Moreover, the algorithm will install a service at a facility if the outgoing edge from the facility node to the service node is on p and if it is not already installed.

Remark. It is worth noting that the solution paths can be found by running *any* online algorithm for the well-known *Online Steiner Forest* problem (OSF). In OSF, we are given an edge-weighted graph and pairs of nodes arriving over time. As soon as a pair arrives, the online algorithm needs to ensure that there is at least one path, whose edges are purchased by the algorithm, that connects the pair. The goal is to minimize the total weights of the edges purchased. The best competitive ratio achievable for OSF is $O(\log^2 n)$, where n is the number of nodes in the graph. There exists an online algorithm in the literature with this ratio, due to Awerbuch *et al.* (Awerbuch et al., 2004). Nevertheless, this algorithm will not imply a desirable asymptotically optimal competitive ratio for our problem, without further analysis. This is because the number of nodes in our formulated graph can get as many as $n+k+mk$, implying a competitive ratio of $O(\log^2(km+n))$ for non-metric OFL-SIC. Hence, a different approach is required.

Random Variable α. Before any request arrives, the algorithm chooses α to be the minimum among $2\lceil\log(kn)\rceil$ independently chosen random variables, distributed uniformly in the interval $[0,1]$.

The algorithm finds the solution paths as follows. It initially knows k, the number of services, and n, the number of requests. Let G be the formulated graph, growing over time as new requests arrive. We assign to each edge e in G of weight w_e, a fraction f_e. We set the values of all fractions to 0 initially. The online algorithm will increase these fractions throughout its execution. We define the *max-flow* from node u to node v in G to be the smallest total fractions of edges which if removed would disconnect u from v. These edges form a *min-cut* from u to v in G.

Whenever a new request arrives, a request node and its outgoing edges are added to G. A fraction set initially to 0 is assigned to each new edge. The *weight of a path* is the sum of the weights of its edges.

For each service requested, the algorithm executes the following. Let r be the request node and s the service node requested.

Online Algorithm for Non-metric OFL-SIC.

1. While the max-flow from the request node r to the service node s in G is less than 1, construct a min-cut K from r to s in G; for each edge $e \in K$, make the following **increment**:

$$f_e = f_e \cdot (1 + 1/w_e) + \frac{1}{|K| \cdot w_e}$$

2. Purchase each edge e with $f_e > \alpha$.
3. If there is no purchased path from r to s in G, find a minimum-weight such path and purchase it.

Notice that the last step guarantees that the algorithm achieves a feasible solution.

6 COMPETITIVE ANALYSIS

In this section, we show that our algorithm has an $O(\log(nk)\log m)$-competitive ratio, where m is the number of facilities, n is the number of requests, and k is the number of services.

The algorithm makes purchases in Step 2 and Step 3. We will measure the cost of the algorithm in each step separately. Note that the cost of the algorithm is equivalent to the total weight of the edges purchased in the formulated graph.

Algorithm's Cost in Step 2. Let E' be the set of the edges purchased in the second step of the algorithm and let $Cost_{E'}$ be the expected cost. Recall that an edge is purchased if its fraction exceeds α. We fix any $i : 1 \leq i \leq 2\lceil\log(kn+1)\rceil$ and an edge e. We denote by $X_{e,i}$ the indicator variable of the event that e is purchased by the algorithm. Let w_e be the weight of edge e and f_e its fraction. We have that:

$$Cost_{E'} = \sum_{e\in E'} \sum_{i=1}^{2\lceil\log(kn+1)\rceil} w_e \cdot Exp[X_{e,i}] = 2\lceil\log(kn+1)\rceil \sum_{e\in E'} w_e f_e \tag{1}$$

Next, we give an upper bound for $\sum_{e\in E'} w_e f_e$ in terms of the optimal offline solution. Let E be the set of all edges in the graph. $\sum_{e\in E} w_e f_e$ (i.e., with all edges included) is the cost of the so-called *fractional solution* of the algorithm. In a fractional solution, the algorithm is allowed, for each edge, to buy a fraction of it and pay the corresponding fraction of its cost. The cost of a fractional solution is called the *fractional cost* of the algorithm. When fractions are either 0 or 1, then the solution is called an *integral solution*. The algorithm's goal is to ultimately find a feasible integral solution. To achieve that, the algorithm produces, in the first step, a fractional solution, which is rounded into an integral solution in the second step. The integral solution in the second step is not necessarily feasible. In the third step, the algorithm ensures that the final integral solution is feasible.

Now, we compare the cost of the fractional solution to the cost of the optimal integral solution. The following lemma will be used in the comparison.

Lemma 1. *Every min-cut constructed in Step 1 contains at least one edge of the optimal integral solution.*

Proof. Assume there is no such edge. Given a pair that needs to be connected. The optimal solution needs to connect this pair through at least one path, p. By the definition of a cut, every cut should contain at least one of the edges of p. □

We call it a *min-cut construction* every time the algorithm constructs a min-cut. Observe that, each *optimal edge*, i.e., an edge in the optimal solution, can appear in zero, or more min-cut constructions, which do not have to be consecutive. We look into all the optimal edges that appeared in at least one min-cut construction. We calculate the costs paid by the fractional solution during all the min-cut constructions in which each of these edges appeared. By doing so, we would have measured the total cost of the fractional solution. This is true because an increment is only made after a min-cut construction and due to Lemma 1.

Let e be an edge that appeared in more than one min-cut construction. The optimal algorithm pays w_e. We calculate now what the online algorithm pays during the min-cut constructions in which e appeared.

Lemma 2. *Each increment increases the fractional cost by at most 2.*

Fix any min-cut K constructed. Each edge e in K increases the cost by $w_e \cdot \left(\frac{f_e}{w_e} + \frac{1}{|K|\cdot w_e}\right)$. Before an increment, the max-flow is less than 1 (or $\sum_{e\in K} f_e < 1$). Hence, adding up over all $|K|$ edges, we get for each increment a total cost of:

$$\sum_{e\in K} w_e \cdot \left(\frac{f_e}{w_e} + \frac{1}{|K|\cdot w_e}\right) < 2$$

The fraction f_e of e will become 1 after a finite number of min-cut constructions and this number can be upper bounded as follows. Based on the increment equation of the algorithm, after $O(w_e \log|K|)$ min-cut constructions, f_e becomes 1 and e cannot appear in any further min-cut construction. Due to Lemma 2 and since each min-cut construction is accompanied with only one increment, we imply that the algorithm pays at most $O(w_e \log|K|)$ during the min-cut constructions in which e appeared. This is $O(\log|K|)$ times what the optimal has paid. Moreover, we have that $|K|$ is at most m, which is the number of paths between any pair, each containing one facility node. The same analysis holds for every optimal edge appearing in at least one min-cut construction. By summing up over all these edges, we achieve an upper bound for the fractional cost of the algorithm:

$$\sum_{e\in E} w_e f_e \leq O(\log m \cdot Opt) \tag{2}$$

Since $\sum_{e\in E'} w_e f_e \leq \sum_{e\in E} w_e f_e$ and from Equations 1 and 2, we conclude that:

$$Cost_{E'} \leq O(\log(nk)\log m \cdot Opt) \tag{3}$$

Algorithm's Cost in Step 3. Now, we measure the cost of the algorithm in the third step. Let $Cost_{E''}$ be the expected cost of this step.

We fix a pair (request node r, service node s) and $1 \leq i \leq 2\lceil\log(kn+1)\rceil$ accompanied with the random variable α chosen before the arrival of requests. We record the time at which the algorithm has already completed the second step. We fix any min-cut K from r to s at this point. The probability that the algorithm did not purchase a path from r to s in the second step is equal to the probability that it did not purchase any of the edges in K. To see why this holds, we let e be an edge in K. All the edges in every path containing e must have a fraction equal to at least the fraction of e (Max flow - Min Cut theorem). So if e is purchased, i.e., $e > \alpha$, then all the other edges on the paths containing e are purchased too, since they have a higher fraction than e. Hence, the probability is equal to:

$$\prod_{e \in K} (1 - f_e) \leq e^{-\sum_{e \in K} f_e} \leq 1/e$$

The second inequality holds since the algorithm ensures that $\sum_{e \in K} f_e \geq 1$ at this point. Thereby, the expected cost of purchasing a path in the third step, for all $1 \leq i \leq 2\lceil \log(kn+1) \rceil$, would be less than $1/(kn)^2 \cdot Opt$, where the optimal solution cost Opt can be used as an upper bound for the minimum-weight path constructed by the algorithm in the third step.

The total number of pairs the algorithm receives is at most kn, since each of the n requests can ask for at most k services. Summing up over all these pairs, we conclude the expected cost of the algorithm in the third step:

$$Cost_{E''} \leq 1/(kn) \cdot Opt \tag{4}$$

By combining Equations 3 and 4, we conclude the following.

Theorem 2. (Upper Bound). *There is an online randomized algorithm for the non-metric Online Facility Location with Service Installation Costs (OFL-SIC), that has an asymptotically optimal competitive ratio of $O(\log(nk)\log m)$, where m is the number of facilities, n is the number of requests, and k is the number of services.*

7 CONCLUDING REMARKS & FUTURE WORK

In this paper, we have studied the non-metric *Online Facility Location with Service Installation Costs* problem (OFL-SIC), which could also be called the non-metric *Online Multi-Commodity Facility Location with linear costs* problem (non-metric OMCFL with linear costs). A next step would be to consider the non-metric *Online Multi-Commodity Facility Location* problem (OMCFL) for other facility cost functions, such as the cost functions defined for the metric case in (Castenow et al., 2020). It seems like other techniques than the ones used in this paper would be needed to achieve results for these cost functions.

Moreover, unlike in the offline setting, for the general facility cost function, there are no online algorithms in the literature for both the metric and non-metric cases. So there is a lot to investigate in this direction.

Another direction is to assume facilities with capacities, for both the metric and non-metric variants. This would reflect a more natural real-world facility location problem, in which the number of clients served by each facility is limited by the resources available at the facility (Cygan et al., 2018).

REFERENCES

Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., and Naor, J. S. (2006). A general approach to online network optimization problems. *ACM Trans. Algorithms*, 2(4):640–660.

Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., and Naor, J. S. (2009). The online set cover problem. *SIAM J. Comput.*, 39(2):361–370.

Awerbuch, B., Azar, Y., and Bartal, Y. (2004). Online generalized steiner problem. *Theoretical Computer Science*, 324(2):313 – 324. Online Algorithms: In Memoriam, Steve Seiden.

Castenow, J., Feldkord, B., Knollmann, T., Malatyali, M., and Meyer Auf der Heide, F. (2020). The online multicommodity facility location problem. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '20, page 129–139, New York, NY, USA. Association for Computing Machinery.

Cygan, M., Czumaj, A., Mucha, M., and Sankowski, P. (2018). Online facility location with deletions. *ArXiv*, abs/1807.03839.

Fleischer, R., Li, J., Tian, S., and Zhu, H. (2006). Non-metric multicommodity and multilevel facility location. In Cheng, S.-W. and Poon, C. K., editors, *Algorithmic Aspects in Information and Management*, pages 138–148, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fotakis, D. (2003). On the competitive ratio for online facility location. In Baeten, J. C. M., Lenstra, J. K., Parrow, J., and Woeginger, G. J., editors, *Automata, Languages and Programming*, pages 637–652, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fotakis, D. (2007). A primal-dual algorithm for online non-uniform facility location. *Journal of Discrete Algorithms*, 5(1):141 – 148.

Korman, S. (2005). On the use of randomization in the online set cover problem. Master's thesis, Weizmann Institute of Science, Israel.

Meyerson, A. (2001). Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431.

Meyerson, A. M. (2005). The parking permit problem. In *Proceedings of the 46th Annual IEEE Symposium on*

Foundations of Computer Science, FOCS '05, page 274–284, USA. IEEE Computer Society.

Poplawski, L. J. and Rajaraman, R. (2011). Multicommodity facility location under group steiner access cost. In *SODA '11*.

Ravi, R. and Sinha, A. (2010). Approximation algorithms for multicommodity facility location problems. *SIAM J. Discret. Math.*, 24(2):538–551.

Shmoys, D. B., Swamy, C., and Levi, R. (2004). Facility location with service installation costs. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, page 1088–1097, USA. Society for Industrial and Applied Mathematics.

Svitkina, Z. and Tardos, E. (2010). Facility location with hierarchical facility costs. *ACM Trans. Algorithms*, 6(2).