Online Algorithms for Leasing Vertex Cover and Leasing Non-metric Facility Location

Christine Markarian[1] and Friedhelm Meyer auf der Heide[2]
[1]Department of Mathematical Sciences, Haigazian University, Beirut, Lebanon
[2]Heinz Nixdorf Institute and Computer Science Department, University of Paderborn, Paderborn, Germany

Keywords: Online Algorithms, Competitive Analysis, Leasing, Parking Permit Problem, Randomized Rounding, Primal-Dual Algorithms, Operations Research, Resource Allocation.

Abstract: We consider leasing variants of two classical $\mathcal{NP}$-hard optimization problems, *Vertex Cover* (VC) and *non-metric Facility Location* (non-metric FL). These contain the well-known *Parking Permit* problem due to Meyerson [FOCS 2005] as a special sub-case and can be found as sub-problems in many operations research applications. We give the first online algorithms for these two problems, evaluated using the standard notion of *competitive analysis* in which the online algorithm whose input instance is revealed over time is compared to the optimal offline algorithm which knows the entire input sequence in advance and is optimal. Our algorithms have optimal and near-optimal competitive ratios for the leasing variants of VC and non-metric FL, respectively.

1 INTRODUCTION

Many companies have stopped buying their resources and started leasing them instead. This has reduced their costs and provided them with flexibility not offered by buying. As a result of this flexibility, more complex decisions had to be made since more options were now available, including different prices for a single resource depending on the length and time of the lease in comparison to a fixed buying price. These options respect *economy of scale* such that a longer lease costs less per unit time. The aim is to decide *which* lease to buy *when* without making regrets in the future, that is, while minimizing costs. Should we know future demands in advance, we would make use of the wide range of offline approaches in the literature that would yield optimal leasing decisions. However, since demands are not always known in advance, leasing decisions are often made *online*. To better grasp the difficulty of such scenarios and make wiser decisions, many theoretical leasing models have been studied, the first of which was by Meyerson in 2005 who introduced a toy example problem, the *Parking Permit* problem (PP) (Meyerson, 2005). Each day, depending on the weather, we have to either use the car (if it is rainy) or walk (if it is sunny). In the former case, we must have a valid parking permit, which we choose among $|\mathcal{L}|$ different lease types. At any time, lease prices respect economy of scale. The goal is to buy a set of leases in order to cover all rainy days while minimizing the total cost of purchases and without using weather forecasts. Interestingly, it turns out that providing a provably good solution to this simple problem requires a challenging algorithmic approach due to the online nature of leasing. To measure the quality of his approach, Meyerson (Meyerson, 2005) used the standard notion of *competitive analysis* that is commonly used to measure the performance of online algorithms (Sleator and Tarjan, 1985). Given an optimization problem X, the worst case of the ratio over all instances of X between the cost incurred by an online algorithm Alg and the optimal offline algorithm is called the *competitive ratio* of Alg. An online algorithm with competitive ratio r is said to be *r-competitive*. Meyerson (Meyerson, 2005) gave a deterministic $O(|\mathcal{L}|)$-competitive and a randomized $O(\log|\mathcal{L}|)$-competitive algorithm along with matching lower bounds for PP. A series of more complex leasing problems generalizing PP followed Meyeron's work. These include the leasing variants of classical $\mathcal{NP}$-hard optimization problems found in various application areas, such as *Set Cover* (Abshoff et al., 2016), *metric Facility Location* (Abshoff et al., 2016; Nagarajan and Williamson, 2013), and *edge-weighted Steiner Tree/ Forest* (Bienkowski et al., 2017; Meyerson, 2005). Unlike in the classical non-

Markarian, C. and der Heide, F.
Online Algorithms for Leasing Vertex Cover and Leasing Non-metric Facility Location.
DOI: 10.5220/0007369503150321
In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES 2019)*, pages 315-321
ISBN: 978-989-758-352-0

leasing setting in which when a resource is chosen into the solution, it can be used forever without incurring future costs, resources in the leasing setting have duration and expire. In order to further use a resource, it needs to be paid for. These leasing variants are natural generalizations that capture more realistic application scenarios. The leasing model by Meyerson has later seen a number of extensions, including lease prices changing over time (Feldkord et al., 2017), leases with length *and* capacity (de Lima et al., 2017a), and leases accompanied with penalties (Markarian, 2017).

In this paper, we continue the algorithmic study of leasing problems by considering the leasing variants of two classical $\mathcal{NP}$-hard optimization problems, *Vertex Cover* and *(non-metric) Facility Location*. As far as we are aware of, no online algorithms are known for them in the literature. In what follows, we define these two variants and state our result for each.

1.1 Leasing Vertex Cover

Definition 1. *(Leasing Vertex Cover (LVC)) We are given an undirected graph $G = (V,E)$ with $|V| = n$ and node-weight function $w : V \to \mathcal{R}^+$. There is a set $\mathcal{L}$ of different lease types each characterized by a duration and cost, and a sequence of edges of G arriving over time. A node can be leased using lease type l for cost c_l and remains* active *for time d_l. In each step t, an edge (u,v) arrives and LVC asks is to ensure that either node u or node v is active at time t, while minimizing overall leasing costs.*

A problem related to LVC is the *Online Vertex Cover* problem (OVC) due to Demange *et al.* (Demange and Paschos, 2005), in which the input graph is not known in advance and the nodes along with their incident edges are revealed over time. The goal is to construct a minimum weight subset S of nodes such that each arriving edge is *covered* by S (i.e., is incident to at least one node in S). Demange *et al.* (Demange and Paschos, 2005) gave upper and lower bounds depending on the maximum degree of the input graph for OVC. Our model in LVC is quite different, since *edges* are revealed over time, such that each edge may appear more than once and needs to be covered only at the current time step it arrives. LVC has deterministic $\Omega(|\mathcal{L}|)$ and randomized $\Omega(\log|\mathcal{L}|)$ lower bounds on its competitive ratio resulting from lower bounds for PP.

Result 1. *We propose an $O(|\mathcal{L}|)$-competitive deterministic algorithm for LVC, which is asymptotically optimal and is based on a simple primal-dual approach.*

1.2 Leasing Non-metric Facility Location

Definition 2. *(Leasing Non-metric Facility Location (LNFL)) We are given an undirected connected graph $G = (V,E)$ with $|V| = n$, node-weight function $w : V \to \mathcal{R}^+$, and unit edge-weight function $w : E \to 1$. There is a set $\mathcal{L}$ of different lease types each characterized by a duration and cost, and a sequence of nodes of G arriving over time. The distance d_{uv} between nodes u and v of G is the total edge-weight of the shortest path between u and v. There are m nodes assigned as potential facilities. A facility node can be leased using lease type l for cost c_l and remains* active *for time d_l. In each step t, a node $u \in V$ (client) arrives and needs to be connected to a node of G (facility) active at time t. Connecting node u to node v costs the distance d_{uv} between them. LNFL asks to connect all arriving nodes, while minimizing overall leasing and connection costs.*

A closely related problem is the *Leasing Metric Facility Location* problem, the metric variant of LNFL to which Nagarajan and Williamson (Nagarajan and Williamson, 2013) gave an $O(|\mathcal{L}| \cdot \log n)$-competitive algorithm. Kling *et al.* (Kling et al., 2012) extended this result by giving an $O(l_K \log(l_K))$-competitive algorithm, where l_K is the maximum lease length. Both of these works exploit triangle inequality in their competitive analysis and hence do not extend to LNFL. Furthermore, LNFL generalizes the *Online Non-metric Facility Location* problem (ONFL) to which Alon *et al.* (Alon et al., 2006) gave a randomized $O(\log n \log m)$-competitive algorithm. The *Online Set Cover* problem (OSC) due to Alon *et al.* (Alon et al., 2003) is a special case of ONFL in which the facilities are sets and the connection cost is 0 if the element belongs to the set and infinity otherwise. OSC has a randomized lower bound of $\Omega(\log m \log n)$ due to Korman (Korman, 2005), where m is the number of sets and n is the number of elements. Hence, LNFL has a randomized lower bound of $\Omega(\log n \log m + \log|\mathcal{L}|)$ on its competitive ratio resulting from lower bounds for OSC and PP.

Result 2. *We propose an $O(\log n \log m + \log|\mathcal{L}| \log n)$-competitive randomized algorithm for LNFL, based on rounding online a fractional solution constructed using a multiplicative incremental approach. The latter is a common technique used in many online covering problems, one of the first of which is OSC due to Alon* et al. *(Alon et al., 2003). Our algorithm is based on the randomized algorithm for ONFL due to Alon* et al. *(Alon et al., 2006), which adopts a similar technique.*

Outline. The rest of the paper is organized as follows. In Section 2, we address the offline variants of LVC and LNFL and give some preliminaries. In Sections 3 and 4, we present the online algorithms for LVC and LNFL, respectively, and evaluate their performance using competitive analysis. We conclude in Section 5 with some open problems.

2 OFFLINE MODEL & INTERVAL MODEL

In this section, we first address the offline variants of LVC and LNFL and then define a simplified model for the lease types, called the *Interval Model*, which we use throughout the paper.

2.1 Offline Model

Koutris (Koutris, 2010) gave a 6-approximation algorithm for the offline variant of LVC, based on a standard primal-dual approach. The latter is an offline algorithm for a generalization of LVC, the *Leasing Set Cover* problem, with $(3d)$-approximation ratio, where d is the maximum number of subsets an element belongs to. As for the offline variant of LNFL, we use the following transformation. An offline instance of *Leasing Non-metric Facility Location* can be transformed into an instance of the offline variant of *Leasing Set Cover*, in which elements are represented by n clients and subsets are formulated by taking all combinations formed by m facilities and $|\mathcal{L}|$ lease types over n time steps. There is an $O(\log n)$ approximation for the offline variant of *Leasing Set Cover* due to Anthony *et al.* (Anthony and Gupta, 2007), based on reducing the latter to its corresponding multistage stochastic optimization problem. Note that a similar transformation between the online variants of LVC and LNFL is possible but yields a competitive ratio that is polynomial in m for LNFL. This can be done by applying the $O(\log n \log(m \cdot |\mathcal{L}|))$-competitive randomized algorithm for *Leasing Set Cover* due to Abschoff *et al.* (Abshoff et al., 2016). Moreover, *Leasing Set Cover* has a lower bound that depends on the number of subsets m, $\Omega(\log n \log m + \log|\mathcal{L}|)$. Hence, no online algorithm for *Leasing Set Cover* will yield a better competitive ratio for LNFL using this transformation.

2.2 Interval Model

We assume the following model, known as the *Interval model* due to Meyerson (Meyerson, 2005), for the leasing options.

Definition 3. *(Interval Model) Leases in the interval model satisfy the following two properties: (i) Lease lengths l_k are powers of two. (ii) Leases of the same type do not overlap.*

Meyerson (Meyerson, 2005) showed that the interval model simplifies the original leasing model at the cost of a constant factor in the competitive ratio. This model has been adopted in all other leasing optimization problems studied thus far (Abshoff et al., 2016; Bienkowski et al., 2017; Nagarajan and Williamson, 2013).

3 LEASING VERTEX COVER

In this section, we present a deterministic online algorithm for LVC and show its competitive analysis.

3.1 Online Algorithm

We formulate LVC as an integer linear program (see Figure 1). Nodes are represented as triplets, such that node $u \in V$ with lease type l at starting time step t is denoted as (u,l,t). The collection of all these triplets is denoted as $\mathcal{V}$. $x_{(u,l,t)}$ is a primal variable indicating whether (u,l,t) is bought (= 1) or not (= 0). c_l is the cost of lease type l and d_l is its duration. w_u is the weight of node u. Edges are represented as pairs, such that edge $(u,v) \in E$ arriving at time t is denoted as (uv,t). Recall that an edge may appear at more than one time step. The collection of all these pairs is denoted as $\mathcal{D}$. The steps of the algorithm upon the arrival of a new edge are depicted in Algorithm 1.

Algorithm 1: Online Deterministic Primal-dual Algorithm for LVC.

When an edge (u,v) arrives at time t,
(i) increase its dual variable $y_{(uv,t)}$ until the dual constraint of a triplet covering time step t corresponding to one of its endpoints becomes tight.
(ii) set the primal variable $x_{(u,l,t)}$ corresponding to every tight dual constraint to 1.

3.2 Competitive Analysis

Algorithm 1 constructs a feasible dual solution, since it never violates the dual constraints - it stops increasing the dual variables as soon as a constraint is tight. Furthermore, since the algorithm makes sure the primal variable of at least one endpoint covering time step t of each edge arriving at time t is set to 1, the primal solution constructed is also feasible. It remains

$$\min \sum_{(u,l,t)\in\mathcal{V}} x_{(u,l,t)} \cdot c_l \cdot w_u$$

$$\text{Subj to: } \forall (uv,t') \in \mathcal{D}: \sum_{\substack{(u,l,t)\\(v,l',t'')\in\mathcal{V}\\t'\in[t,t+d_l]\\t'\in[t'',t''+d_{l'}]}} x_{(u,l,t)} + x_{(v,l',t'')} \geq 1$$

$$\forall (u,l,t) \in \mathcal{V}: x_{(u,l,t)} \in \{0,1\}$$

$$\max \sum_{(uv,t')\in D} y_{(uv,t')}$$

$$\text{Subj to: } \forall (u,l,t) \in \mathcal{V}: \sum_{\substack{v\in V\\(uv,t')\in\mathcal{D}\\t'\in[t,t+d_l]}} y_{(uv,t')} \leq c_l \cdot w_u$$

$$\forall (uv,t') \in \mathcal{D}: y_{(uv,t')} \geq 0$$

Figure 1: ILP Formulation of Leasing Vertex Cover

to show a relationship between the cost of the primal solution and that of the dual solution. Let $P \subseteq \mathcal{V}$ denote the primal solution constructed by the algorithm. Since the dual constraint is tight for each $(u,l,t) \in P$, we have

$$c_l \cdot w_u = \sum_{\substack{v\in V\\(uv,t')\in\mathcal{D}\\t'\in[t,t+d_l]}} y_{(uv,t')}$$

Summing up over all triplets in P, we get

$$\sum_{(u,l,t)\in P} c_l \cdot w_u = \sum_{(u,l,t)\in P} \sum_{\substack{v\in V\\(uv,t')\in\mathcal{D}\\t'\in[t,t+d_l]}} y_{(uv,t')}$$

$$= \sum_{(uv,t')\in\mathcal{D}} y_{(uv,t')} \sum_{\substack{(u,l,t)\in P\\(v,l',t'')\in P\\t'\in[t,t+d_l]\\t'\in[t'',t''+d_{l'}]}} 1$$

Assuming the interval model, we have that on each day t, there are exactly $|\mathcal{L}|$ leases covering time step t. Since each edge can be covered by at most two nodes, the algorithm can only set the primal variables corresponding to at most $2 \cdot |\mathcal{L}|$ triplets to 1. Thus,

$$\sum_{\substack{(u,l,t)\in P\\(v,l',t'')\in P\\t'\in[t,t+d_l]\\t'\in[t'',t''+d_{l'}]}} 1 \leq 2 \cdot |\mathcal{L}|$$

By Weak Duality theorem, we have that any feasible dual solution is a lower bound to any feasible primal solution, including the optimal primal solution. This completes the analysis and the theorem below follows.

Theorem 1. *There is an $O(|\mathcal{L})|$-competitive deterministic algorithm for Leasing Vertex Cover.*

4 LEASING NON-METRIC FACILITY LOCATION

In this section, we present a randomized online algorithm for LNFL and show its competitive analysis.

4.1 Online Algorithm

We formulate LNFL as follows. We are given a bipartite graph $\mathcal{G} = (A \cup B, \mathcal{E})$, where A contains the set of n nodes and B contains the set of m potential facilities of the input graph G, and a root node r. Nodes have no weights in $\mathcal{G}$. The edge set $\mathcal{E}$ is weighted and formed as follows. There is an edge between each node in A and each node in B. An edge (u,v) between node $u \in A$ and node $v \in B$ has cost equal to the distance d_{uv} between nodes u and v in G (cost of edge (u,u) is set to 0). There are $|\mathcal{L}|$ edges between each node u in B and root node r, corresponding to each of the lease types. Leases are structured according to the Interval model described earlier. There is a cost associated with edge (u,r) of lease type l, equal to $(c_l \cdot w_u)$, where c_l is the cost of lease type l and w_u is the weight of node u in G. In each step, a node from A arrives and needs to be connected to r. The online algorithm needs to ensure that there is a path between each arriving node to r at the time of arrival, while minimizing the total costs.

We denote by c_e the cost of edge e. The algorithm assigns to each edge e of $\mathcal{G}$ associated with a time interval and lease type, a fraction f_e, initially set to 0. We define the *maximum flow* between two nodes u and v to be the smallest total *fraction* of edges which if removed would disconnect u from v. These edges form a *minimum cut*. When a node arrives at time t, the algorithm disregards:

- all edges whose lease does not cover time t
- all edges between nodes in A that previously arrived and all nodes in B w

A random variable μ is chosen as the minimum among $2\lceil\log(n+1)\rceil$ independent random variables, distributed uniformly in the interval $[0,1]$. All logarithms are to the base 2. We say a node u arriving at time t is *connected* to r if there is a path of *leased* edges covering step t from u to r. The steps of the algorithm upon the arrival of a new node are depicted in Algorithm 2 below.

4.2 Competitive Analysis

Algorithm 2 ensures a feasible integral solution upon its termination, due to Step (iii). It also constructs a feasible fractional solution in Step (i). We manage to

Algorithm 2: Online Randomized Algorithm for LNFL.

Input. Graph $\mathcal{G}$ and node u arriving at time t
Output. Set of leased edges covering step t
When u arrives,
(i) If u is connected to r, do nothing.
Else, while the maximum flow between u and r is less than 1,
- compute a minimum cut $\mathcal{C}$ between u and r
- increment the fraction f_e of each edge $e \in \mathcal{C}$ with the equation:

$$f_e \leftarrow f_e(1+\frac{1}{c_e}) + \frac{1}{|\mathcal{C}| \cdot c_e}$$

(ii) Lease edge e if $w_e \geq \mu$.
(iii) If u is not connected to r, lease the edges of a path from u to r of cheapest cost.

bound the cost of the fractional solution constructed in terms of the cost of the optimal integral solution (Lemma 1). We also show that the cost of our integral solution constructed can be bounded in terms of the cost of our fractional solution (Lemma 2) and conclude the competitive ratio of the algorithm.

Let $\mathcal{E}'$ be the collection of all leases the algorithm can purchase, that is, all edges associated with all time intervals and all lease types. We refer to edges in $\mathcal{E}'$ as *edge-intervals*.

Lemma 1. *The cost of the algorithm's fractional solution C_{frac} is at most $O(\log(m \cdot |\mathcal{L}|)) \cdot Opt$, where Opt is the cost of the optimal integral solution.*

Proof. The fractional solution constructed has cost:

$$\sum_{e \in \mathcal{E}'} c_e \cdot f_e$$

Consider an iteration in which the algorithm is given a node u that is not connected to r. We show that every increment the algorithm performs can be upper bounded by 2. Notice that the algorithm performs an increment only if the maximum flow is less than 1. Hence the edges of every minimum cut have a total fraction of less 1 before an increment. Let $\mathcal{C}$ be a minimum cut between u and r. The total fractional cost added by the edge-intervals of $\mathcal{C}$ is at most:

$$\sum_{e \in \mathcal{C}} c_e \cdot (\frac{f_e}{c_e} + \frac{1}{|\mathcal{C}| \cdot c_e}) \leq 2$$

Now we need an upper bound on the total number of increments the algorithm performs. Any optimal integral solution must contain at least one of the edge-intervals of each minimum cut, by the definition of minimum cut. Let us fix such an edge-interval e' in a minimum cut $\mathcal{C}_{e'}$. The total number of increments needed to make the fraction $f_{e'}$ 1 can be upper bounded by $(c_{e'} \cdot \log|\mathcal{C}_{e'}|)$. As soon as this fraction gets 1, its edge-interval can't be in any future minimum cut. Adding up over the set $\mathcal{E}'_{opt} \subseteq \mathcal{E}'$ of all edge-intervals of the optimal integral solution, we get:

$$\sum_{e' \in \mathcal{E}'_{opt}} c_{e'} \cdot \log|\mathcal{C}_{e'}|$$

Now we upper bound the number of edge-intervals in any minimum cut by $(m \cdot |\mathcal{L}|)$. Fix a facility node $v \in B$. A minimum cut between a node u to r through v will either contain the edge-interval corresponding to edge (u,v) or the $|\mathcal{L}|$ edge-intervals corresponding to the edges from v to r. Since there are m edge-intervals from u to all other facility nodes and each of the m facility nodes has $|\mathcal{L}|$ edge-intervals to r, t $|\mathcal{C}_{e'}| \leq (m \cdot |\mathcal{L}|)$ for all $e' \in \mathcal{E}'_{opt}$. Adding up over all edge-intervals of the optimal integral solution, we get:

$$\sum_{e \in \mathcal{E}'} c_e \cdot f_e \leq O(\log(m \cdot |\mathcal{L}|)) \cdot Opt$$

□

Lemma 2. *The cost of the algorithm's integral solution is at most $O(\log n) \cdot C_{frac}$, where C_{frac} is the cost of the algorithm's fractional solution.*

Proof. The algorithm purchases its edge-intervals in Steps (ii) and (iii) only. We measure the expected cost of the integral solution in each step. Let $X_e(i)$ be the random variable indicating the event that $f_e > \mu$ for $1 \leq i \leq 2\lceil\log(n+1)\rceil$. The expected cost $E(\sum_{e\in\mathcal{E}'} c_e)$ of all edge-intervals is at most:

$$\sum_{e \in \mathcal{E}'} \sum_{i=1}^{2\lceil\log(n+1)\rceil} c_e \cdot E(X_e(i)) = \sum_{e \in \mathcal{E}'} \sum_{i=1}^{2\lceil\log(n+1)\rceil} c_e \cdot f_e$$

$$\leq 2\lceil\log(n+1)\rceil \cdot C_{frac}$$

This bounds the integral cost of the algorithm in Step (ii). As for Step (iii), the algorithm performs it only if Step (ii) fails to connect the given node. Fix a node u that was not connected to r in Step (ii). The *flow* of a path is the minimum of all weights of the edge-integrals of the path. At the end of Step (i), the algorithm ensures that the sum of flows of all paths between u and r is at least 1. Hence, the probability that the algorithm fails to connect node u is upper bounded by the probability that μ exceeds the flow of each path from u to r. Fix a minimum cut $\mathcal{C}$ constructed at the end of Step (i). The probability that u is not connected is thus:

$$\prod_{e \in C}(1 - f_e) \leq e^{-\sum_{e \in C} f_e} \leq \frac{1}{e}$$

Hence, for all $1 \leq i \leq 2\lceil \log(n+1) \rceil$, the probability that u is not connected is at most $\frac{1}{n^2}$. In Step (iii), the algorithm purchases edge-intervals of a cheapest path from u to r, which is a lower bound on the cost of the optimal solution Opt. This yields a negligible expected cost of $n \cdot \frac{Opt}{n^2}$, for the at most n arriving nodes, and concludes the proof. □

Lemma 1 and Lemma 2 ultimately lead to the following theorem.

Theorem 2. *There is an $O(\log n \log m + \log |\mathcal{L}| \log n)$-competitive randomized algorithm for Leasing Non-metric Facility Location.*

5 DISCUSSION & OPEN PROBLEMS

We have presented the first online algorithms, with optimal/near-optimal competitive ratios, for the leasing variants of two classical optimization problems. Below is a summary of currently known results as well as our results (in bold) for LVC and LNFL, respectively.

LVC

- Online Lower Bound: $\Omega(|\mathcal{L}|)$ (Meyerson, 2005) (deterministic)
- Offline Lower Bound: 2 (Khot and Regev, 2008) (deterministic)
- **Online Upper Bound: $O(|\mathcal{L}|)$-competitive (deterministic)**
- Offline Upper Bound: 6-approximation (Koutris, 2010) (deterministic)

LNFL

- Online Lower Bound: $\Omega(\log n \log m + \log |\mathcal{L}|)$ (Korman, 2005; Meyerson, 2005) (randomized)
- Offline Lower Bound: $\Omega(\log n)$ (Slavik, 1997) (deterministic)
- **Online Upper Bound: $O(\log n \log(m \cdot |\mathcal{L}|))$-competitive (randomized)**
- Offline Upper Bound: $O(\log n)$-approximation (Anthony and Gupta, 2007) (deterministic)

An interesting open problem is to close the gap between the offline lower and upper bounds for LVC. Another direction is to consider a deterministic algorithm for LNFL. While it seems hard to design a deterministic algorithm with a non-trivial competitive ratio for the latter since the algorithm for the online non-leasing variant of the problem as well those for most algorithms for related problems are randomized, it remains an interesting open question to figure whether it is possible at all.

Furthermore, most online (leasing) problems for variants of *Facility Location* assume an underlying metric and use the associated metric properties in the design and/or analysis of the algorithms (de Lima et al., 2017b; Felice et al., 2015). It would be interesting to look at some of these variants such as *Connected Facility Location*, in the online (leasing) model, without considering an underlying metric.

REFERENCES

Abshoff, S., Kling, P., Markarian, C., Meyer auf der Heide, F., and Pietrzyk, P. (2016). Towards the price of leasing online. *J. Comb. Optim.*, 32(4):1197–1216.

Alon, N., Awerbuch, B., and Azar, Y. (2003). The online set cover problem. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 100–105, New York, NY, USA. ACM.

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.

Anthony, B. M. and Gupta, A. (2007). Infrastructure leasing problems. In Fischetti, M. and Williamson, D. P., editors, *Integer Programming and Combinatorial Optimization*, pages 424–438, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bienkowski, M., Kraska, A., and Schmidt, P. (2017). A deterministic algorithm for online steiner tree leasing. In Ellen, F., Kolokolova, A., and Sack, J.-R., editors, *Algorithms and Data Structures*, pages 169–180, Cham. Springer International Publishing.

de Lima, M., Felice, M. S., and Lee, O. (2017a). On generalizations of the parking permit problem and network leasing problems. *Electronic Notes in Discrete Mathematics*, 62:225 – 230. LAGOS'17 IX Latin and American Algorithms, Graphs and Optimization.

de Lima, M. S., Felice, M. C. S., and Lee, O. (2017b). Connected facility leasing problems. In *Joint Proceedings of the 18th Italian Conference on Theoretical Computer Science and the 32nd Italian Conference on Computational Logic co-located with the 2017 IEEE International Workshop on Measurements and Networking (2017 IEEE M&N), Naples, Italy, September 26-28, 2017.*, pages 162–173.

Demange, M. and Paschos, V. T. (2005). Online vertex-covering. *Theoretical Computer Science*, 332(1):83 – 108.

Feldkord, B., Markarian, C., and Meyer auf der Heide, F. (2017). Price fluctuation in online leasing. In *Combinatorial Optimization and Applications - 11th International Conference, COCOA 2017, Shanghai, China, December 16-18, 2017, Proceedings, Part II*, pages 17–31.

Felice, M. C. S., Cheung, S., Lee, O., and Williamson, D. P. (2015). The online prize-collecting facility location problem. *Electronic Notes in Discrete Mathematics*, 50:151–156.

Khot, S. and Regev, O. (2008). Vertex cover might be hard to approximate to within 2 - epsilon. *Journal of Computer and System Sciences*, 74(3):335 – 349.

Kling, P., Meyer auf der Heide, F., and Pietrzyk, P. (2012). An algorithm for online facility leasing. In *Structural Information and Communication Complexity - 19th International Colloquium, SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers*, pages 61–72.

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

Koutris, P. (2010). Infrastructure Leasing Problems. Master's thesis, University of Athens, Greece.

Markarian, C. (2017). Leasing with uncertainty. In *Operations Research Proceedings 2017, Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Freie Universiät Berlin, Germany, September 6-8, 2017.*, pages 429–434.

Meyerson, A. (2005). The parking permit problem. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 274–284.

Nagarajan, C. and Williamson, D. P. (2013). Offline and online facility leasing. *Discrete Optimization*, 10(4):361–370.

Slavik, P. (1997). A tight analysis of the greedy algorithm for set cover. *Journal of Algorithms*, 25(2):237 – 254.

Sleator, D. D. and Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208.