# Online Deterministic Algorithms for Connected Dominating Set & Set Cover Leasing Problems

Christine Markarian[1] and Abdul-Nasser Kassar[2]
[1]Department of Mathematical Sciences, Haigazian University, Beirut, Lebanon
[2]Department of Information Technology and Operations Management, Lebanese American University, Beirut, Lebanon

Keywords: Connected Dominating Sets, Set Cover, Leasing, Online Algorithms, Competitive Analysis.

Abstract: Connected Dominating Set (CDS) and Set Cover (SC) are classical optimization problems that have been widely studied in both theory and practice, as many variants and in different settings, motivated by applications in wireless and social networks. In this paper, we consider the online setting, in which the input sequence arrives in portions over time and the so-called online algorithm needs to react to each portion. Online algorithms are measured using the notion of competitive analysis. An online algorithm $A$ is said to have competitive ratio $r$, where $r$ is the worst-case ratio, over all possible instances of a given minimization problem, of the solution constructed by $A$ to the solution constructed by an offline optimal algorithm that knows the entire input sequence in advance. Online Connected Dominating Set (OCDS) (Hamann et al., 2018) is an online variant of CDS that is currently solved by a randomized online algorithm with optimal competitive ratio. We present in this paper the first deterministic online algorithm for OCDS, with optimal competitive ratio. We further introduce generalizations of OCDS, in the leasing model (Meyerson, 2005) and in the multiple hop model (Coelho et al., 2017), and design deterministic online algorithms for each of these generalizations. We also propose the first deterministic online algorithm for the leasing variant of SC (Abshoff et al., 2016), that is currently solved by a randomized online algorithm.



## 1 INTRODUCTION

Dominating Set problems, where the goal is to find a minimum subgraph of a given (undirected) graph such that each node is either in the subgraph or has an adjacent node in it, form a fundamental class of optimization problems that have received significant attention in the last decades. The *Connected Dominating Set* problem (CDS) - which asks for a minimum such subgraph that is connected - is one of the most well-studied problems in this class (Du and Wan, 2013) with a wide range of applications in wireless networks (Yu et al., 2013) and social networks (Daliri Khomami et al., 2018; Barman et al., 2018; Halawi et al., 2018; Wagner et al., 2017). CDS is known to be $\mathcal{NP}$-complete even in planar graphs (Garey and Johnson, 1979) and admits an $O(\ln\Delta)$-approximation for general graphs, where $\Delta$ is the maximum node degree of the input graph (Guha and Khuller, 1998). The latter is the best possible unless $\mathcal{NP} \subseteq DTIME(n^{\log\log n})$ (Feige, 1998; Lund and Yannakakis, 1994). Motivated by applications in modern robotic warehouses (D'Andrea, 2012), an online variant of CDS, the *Online Connected Dominating Set* problem (OCDS), has been introduced by Hamann *et al.* (Hamann et al., 2018) - the input to the so-called *online algorithm* is an undirected connected graph $G = (V,E)$, and a sequence of subsets of $V$ arriving over time. OCDS asks to construct a subset $S$ of $V$ inducing a connected subgraph in $G$, such that for each subset $D_t$ of $V$ arriving at time $t$, each node of $D_t$ must be either in $S$ or have an adjacent node in $S$ at time $t$. The goal is to minimize the cardinality of $S$. Online algorithms are evaluated using the notion of *competitive analysis*, in which the performance of the online algorithm is measured against the optimal offline solution. Given an input sequence $\sigma$ - let $\mathcal{C}_A(\sigma)$ and $\mathcal{C}_{OPT}(\sigma)$ be the cost of an online algorithm $A$ and an optimal offline algorithm, respectively. $A$ is said to be $c$-*competitive* (or have competitive ratio $c$) if there exists a constant $\alpha$ such that $\mathcal{C}_A(\sigma) \leq c \cdot \mathcal{C}_{OPT}(\sigma) + \alpha$ for all input sequences $\sigma$. Hamann *et al.* (Hamann et al., 2018) proposed an online randomized algorithm for OCDS, with an asymptotically optimal $O(\log^2 n)$-competitive ratio, where $n$ is the number of nodes.

In this paper, we give the first deterministic



Markarian, C. and Kassar, A.
Online Deterministic Algorithms for Connected Dominating Set Set Cover Leasing Problems.
DOI: 10.5220/0008866701210128
In *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems (ICORES 2020)*, pages 121-128
ISBN: 978-989-758-396-4; ISSN: 2184-4372

algorithm for OCDS, with asymptotically optimal $O(\log^2 n)$-competitive ratio. Moreover, motivated by influence spreading applications in social networks (Berman and Coulston, 1997; Daliri Khomami et al., 2018), in which a small group of people (influential people) is selected to spread information to the rest of the group (dominated people), we study the online variant of the *r-hop Connected Dominating Set* problem, where $r$ is a positive integer that denotes the maximum allowable distance (number of edges or hops) between the influential node and the dominated node. Only offline model for $r$-hop connected dominating sets has been known (Coelho et al., 2017). In our online model, groups of people to be dominated are revealed over time and need to be influenced, rather than all at once as in the offline model.

Many classical optimization problems, including *Set Cover* (Abshoff et al., 2016), *Facility Location* (Nagarajan and Williamson, 2013; Markarian and Meyer auf der Heide, 2019), and *Steiner Tree* (Meyerson, 2005; Bienkowski et al., 2017), have been studied in the online leasing model (Meyerson, 2005) and its extensions (Feldkord et al., 2017), in which rather than being bought, resources are leased for different time duration with costs respecting economy of scale, where a long expensive lease costs less per unit time. In this paper, we give the first deterministic online algorithm for the *Online Set Cover Leasing* problem (OCSL), the leasing variant of *Set Cover* (SC). Given a universe $\mathcal{U}$ and a collection $\mathcal{S}$ of subsets of $\mathcal{U}$, SC asks to find a minimum number of subsets $\mathcal{C} \subseteq \mathcal{S}$ whose union is $\mathcal{U}$. Abshoff *et al.* (Abshoff et al., 2016) gave the first online algorithm for OCSL, which was randomized. Furthermore, we introduce the leasing variants of *Connected Dominating Set* and *r-hop Connected Dominating Set* and give a deterministic algorithm for each. All of our algorithms in this paper are online, deterministic, and evaluated using the standard competitive analysis. Our results are summarized as follows.

- We propose the first deterministic algorithm for the *Online Connected Dominating Set* problem (OCDS), with asymptotically optimal competitive ratio of $O(\log^2 n)$, where $n$ is the number of nodes (Section 3). The currently best result for OCDS is a randomized algorithm by Hamann *et al.* (Hamann et al., 2018), with asymptotically optimal $O(\log^2 n)$-competitive ratio.
- We introduce the *Online r-hop Connected Dominating Set* problem (r-hop OCDS), and give a deterministic $O(2r \cdot \log^3 n)$-competitive algorithm, where $n$ is the number of nodes (Section 4). r-hop OCDS has been studied in the offline setting - Coelho *et al.* (Coelho et al., 2017) gave inapproximability results for the problem in some special graph classes.
- We propose the first deterministic algorithm for the *Online Set Cover Leasing problem* (OSCL), with $O(\log \sigma \log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive ratio, where $m$ is the number of subsets, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length (Section 5). The currently best result for OSCL is a randomized algorithm by Abshoff *et al.* (Abshoff et al., 2016), with $O(\log \sigma \log(m\mathcal{L}))$-competitive ratio.
- We introduce the *Online Connected Dominating Set Leasing* problem (OCDSL), and give a deterministic $O\Big((\sigma+1) \cdot \log \sigma \log(n\mathcal{L} + 2n\frac{\sigma}{l_1}) + \mathcal{L} \cdot \log n\Big)$-competitive algorithm, where $n$ is the number of nodes, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length (Section 6).
- We introduce the *Online r-hop Connected Dominating Set Leasing* problem (r-hop OCDSL), and give a deterministic $O\Big(\mathcal{L}(1+\sigma(2r-1)) \log \sigma \log(n\mathcal{L} + 2n\frac{\sigma}{l_1}) \log n\Big)$-competitive algorithm, where $n$ is the number of nodes, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length (Section 7).

## 2 RELATED WORK

**Online Connected Dominating Sets and Online Set Cover.** While there are many works that address Connected Dominating Set problems and other related problems in the offline setting (Guha and Khuller, 1998; Yu et al., 2013; Haraty et al., 2015; Haraty et al., 2016), only few consider the online setting. Boyar *et al.* (Boyar et al., 2016) studied an online variant of the *Connected Dominating Set* problem (CDS), in which the input graph is unknown in advance, and restricted to a tree, a unit disk graph, or a bounded degree graph. Each step a node is either inserted or deleted and the goal is to maintain a connected dominating set of minimum cardinality. Boyar *et al.* showed that a simple greedy approach attains a $(1+\frac{1}{\mathcal{OPT}})$-competitive ratio in trees - where $\mathcal{OPT}$ is the cost of the optimal offline solution, an $(8+\varepsilon)$-competitive ratio in unit disk graphs - for arbitrary small $\varepsilon > 0$, and $b$-competitive ratio in $b$-bounded degree graphs. Recently, Hamann *et al.* (Hamann et al., 2018) introduced the *Online Connected Dominating Set* problem (OCDS), an online variant of CDS, in which the graph is known in advance, and proposed an $O(\log^2 n)$-competitive randomized algorithm for OCDS in general graphs, where $n$ is the number of nodes. Their work was motivated by applications in modern robotic warehouses, in which geometric graphs were used to model the topology of a warehouse. Alon *et al.* (Alon et al., 2003) gave a deterministic $O(\log m \log n)$-competitive algorithm and an $\Omega(\log m \log n/(\log\log m + \log\log n))$ lower bound for the online variant of the *Set Cover* problem, where $m$ is the number of sets and $n$ is the number of elements. Korman (Korman, 2005) then improved the lower bound to $\Omega(\log m \log n)$. For the unweighted case where costs are uniform, Alon *et al.* (Alon et al., 2003) gave an $O(\log n \log d)$ competitive ratio, which was later improved by Buchbinder *et al.* (Buchbinder and Naor, 2005) to $O(\log(n/Opt)\log d)$, where $Opt$ is the optimal offline solution and $d$ is the maximum number of sets an element belongs to.

**Leasing Variants.** Meyerson (Meyerson, 2005) gave deterministic $O(\mathcal{L})$-competitive and randomized $O(\log \mathcal{L})$-competitive algorithms along with matching lower bounds for the *Parking Permit* problem. He also introduced the leasing variant of the *Steiner Forest* problem, for which he proposed a randomized $O(\log n \log \mathcal{L})$ competitive algorithm, where $n$ is the number of nodes, and $\mathcal{L}$ is the number of lease types. Nagarajan and Williamson (Nagarajan and Williamson, 2013) gave an $O(\mathcal{L} \cdot \log n)$-competitive algorithm for the leasing variant of the *Facility Location* problem, where $n$ is the number of clients. Abshoff *et al.* (Abshoff et al., 2016) gave an online randomized algorithm for the leasing variant of *Set Cover*, with $O(\log(m\mathcal{L})\log\sigma)$-competitive ratio and improved previous results for other online variants of *Set Cover*. Bienkowski *et al.* (Bienkowski et al., 2017) proposed a deterministic algorithm that has an $O(\mathcal{L}\log k)$-competitive ratio for the leasing variant of *Steiner Tree*, where $k$ is the number of terminals.

# 3 ONLINE CONNECTED DOMINATING SET (OCDS)

**Definition.** Given a connected graph $G = (V, E)$ and a sequence of disjoint subsets of $V$ arriving over time. A subset $S$ of $V$ serves as a connected dominating set of a given subset $D$ of $V$ if every node in $D$ is either in $S$ or has an adjacent node in $S$, and the subgraph induced by $S$ is connected in $G$. Each step, a subset of $V$ arrives and needs to be served by a connected dominating set of $G$. OCDS asks to grow a connected dominating set of minimum number of nodes.

**Preliminaries.** A *dominating set* of a subset $D$ is a subset $DS$ of nodes such that each node in $D$ is either in $DS$ or has an adjacent node in $DS$. $DS$ is *minimal* if no proper subset of $DS$ is a dominating set of $D$. A minimal dominating set can be constructed online using the online deterministic algorithm by Alon *et al.* for the *Online Set Cover* problem (OSC) (Alon et al., 2003), the online variant of the classical *Set Cover* problem. A *Set Cover* instance is formed by making each node an element, and corresponding each node to a set that contains the node itself, along with its adjacent nodes. Alon *et al.* (Alon et al., 2003) gave a deterministic $O(\log m \log n)$-competitive algorithm for OSC, where $m$ is the number of sets and $n$ is the number of elements.

A *Steiner tree* of a subset $D$ is a tree connecting each node in $D$ to a given root $s$. A Steiner tree can be constructed online using the online deterministic $O(\log n)$-competitive algorithm by Berman *et al.* (Berman and Coulston, 1997). The *Steiner tree* problem studied by Berman *et al.* (Berman and Coulston, 1997) is for edge-weighted graphs and the algorithmic cost is measured by adding the costs of all edges outputted by the online algorithm. Our model in this paper assumes no weights on the nodes, and hence the competitive ratio given by Berman *et al.* for edge-weighted graphs carries over to our graph model in this paper. To see this, assume we are given a graph $G$ with a weight of 1 on all edges and all nodes, and a set of terminals that need to be connected. Let $Opt_e$ be the cost of an optimal Steiner tree $T$ measured by counting the edges in $T$. Let $Opt_n$ be the cost of an optimal Steiner tree $T'$ measured by counting the nodes in $T'$. We have that $Opt_e = Opt_n + 1$. The proof is straightforward, by contradiction. Assume $Opt_e > Opt_n + 1$. We can construct a tree which has an edge cost lower than that of $T$: the tree $T'$ with edge cost $Opt_n + 1$, and this contradicts the fact that $T$ is an optimal Steiner tree. Now assume $Opt_e < Opt_n + 1$. We can construct a tree which has a node cost lower than that of $T'$: the tree $T$ with node cost $Opt_e - 1$, and this contradicts the fact that $T'$ is an optimal Steiner tree. This would not have been the case had there been non-uniform weights on the nodes since the node-weighted variant of the *Steiner tree* problem generalizes the edge-weighted variant by replacing each edge by a node with the corresponding edge cost. Moreover, the node-weighted variant of the *Steiner tree* problem generalizes the *Online Set Cover* problem which has a lower bound of $\Omega(\log m \log n)$ (Korman, 2005) on its competitive ratio.

**Algorithm.** The algorithm assigns, at the first time step, any of the nodes purchased by the algorithm, as a root node $s$. At time step $t$:

**Input:** $G = (V,E)$, subset $D_t$ of $V$
**Output:** A connected dominating set $CDS_t$ of $D_t$

1. Find a minimal dominating set $DS_t$ of $D_t$.
2. Assign to each node in $DS_t$ a *connecting* node, that is any adjacent node from the set $D_t$. If $t = 1$, assign any of the nodes in $DS_t$ as a root node $s$.
3. Find a Steiner tree that connects all connecting nodes to $s$. Add all the nodes in this tree including the nodes in $DS_t$ and their connecting nodes to $CDS_t$.

**Competitive Analysis.** OCDS has a lower bound of $\Omega(\log^2 n)$, where $n$ is the number of nodes, resulting from Korman's lower bound of $\Omega(\log m \log n)$ for OSC (Korman, 2005), where $m$ is the number of subsets and $n$ is the number of elements.

Let $Opt$ be the cost of an optimal solution $Opt_I$ of an instance $I$ of OCDS. Let $C1$, $C2$, and $C3$ be the cost of the algorithm in the three steps, respectively. The first step of the algorithm constructs online a minimal dominating set. Let $Opt_{DS}$ be the cost of a minimum dominating set of $I$. Note that $Opt_I$ is a dominating set of $I$. Hence, Alon *et al.*'s (Alon et al., 2003) deterministic algorithm yields: $C1 \leq \log^2 n \cdot Opt_{DS} \leq \log^2 n \cdot Opt$. The second step adds at most one node for each node bought in the first step. Hence we have that: $C2 \leq C1$. As for the third step, $Opt_I$ is a Steiner tree for the connecting nodes bought in the second step, since all connecting nodes belong to the set of nodes that need to be served and $Opt_I$ serves as a connected dominating set of these nodes. Let $Opt_{St}$ be the cost of a minimum Steiner tree of these connecting nodes. Since Berman *et al.*'s (Berman and Coulston, 1997) algorithm has an $O(\log n)$-competitive ratio, we conclude that $C3 \leq \log n \cdot Opt_{St} \leq \log n \cdot Opt$. The total cost of the algorithm is then upper bounded by: $C1 + C2 + C3 = (2 \cdot \log^2 n + \log n) \cdot Opt$ and the theorem below follows.

**Theorem 1.** *There is an asymptotically optimal $O(\log^2 n)$-competitive deterministic algorithm for the Online Connected Dominating Set problem, where n is the number of nodes.*

# 4 ONLINE r-hop CONNECTED DOMINATING SET (r-hop OCDS)

**Definition.** Given a connected graph $G = (V,E)$, a positive integer $r$, and a sequence of disjoint subsets of $V$ arriving over time. A subset $S$ of $V$ serves as an $r$-hop connected dominating set of a given subset $D$ of $V$ if for every node $v$ in $D$, there is a vertex $u$ in $S$ such that there are at most $r$ hops (edges) between $v$ and $u$ in $G$, and the subgraph induced by $S$ is connected in $G$. Each step, a subset of $V$ arrives and needs to be served by an $r$-hop connected dominating set of $G$. r-hop OCDS asks to grow an $r$-hop connected dominating set of minimum number of nodes.

OCDS is equivalent to r-hop OCDS with $r = 1$.

**Preliminaries.** Given a graph $G = (V,E)$ and a positive integer $r$. A subset $DS$ of $V$ is an *r-hop dominating set* of a given subset $D$ of $V$ if for every node $v$ in $D$, there is a vertex $u$ in $DS$ such that there are at most $r$ hops between $v$ and $u$ in $G$. $DS$ is *minimal* if no proper subset of $DS$ is an *r-hop dominating set* of $D$. We can transform an *r-hop dominating set* instance into a *Set Cover* instance by making each node an element, and corresponding each node to a set that contains the node itself, along with all nodes that are at most $r$ hops away from it. Hence, we can construct a minimal *r-hop dominating set* by running the online deterministic algorithm by Alon *et al.* for the *Online Set Cover* problem (OSC) (Alon et al., 2003). A Steiner tree can be constructed online, as in Section 3, using the online deterministic $O(\log n)$-competitive algorithm by Berman *et al.* (Berman and Coulston, 1997).

**Algorithm.** The algorithm assigns, at the first time step, any of the nodes purchased by the algorithm, as a root node $s$. At time step $t$:

**Input:** $G = (V,E)$, subset $D_t$ of $V$
**Output:** An $r$-hop connected dominating set $rCDS_t$ of $D_t$

1. Find a minimal $r$-hop dominating set $rDS_t$ of $D_t$. If $t = 1$, assign any of the nodes in $rDS_t$ as a root node $s$.
2. Find a Steiner tree that connects all nodes in $rDS_t$ to $s$. Add all the nodes in this tree including the nodes in $rDS_t$ to $rCDS_t$.

**Competitive Analysis.** The only lower bound for r-hop OCDS is the one for OCDS, $\Omega(\log^2 n)$, where $n$ is the number of nodes. The proof of the competitive analysis is ommited due to lack of space.

**Theorem 2.** *There is a deterministic $O(2r \cdot \log^3 n)$-competitive algorithm for the Online r-hop Connected Dominating Set problem, where n is the number of nodes.*

## 5 ONLINE SET COVER LEASING (OSCL)

**Definition.** Given a universe $\mathcal{U}$ of elements ($|\mathcal{U}| = n$), a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ ($|\mathcal{S}| = m$), and a set of $\mathcal{L}$ different lease types, each characterized by a duration and cost. A subset can be leased using lease type $l$ for cost $c_l$ and remains active for $d_l$ time steps. Each time step $t$, an element $e \in \mathcal{U}$ arrives and there needs to be a subset $S \in \mathcal{S}$ active at time $t$ such that $e \in S$. OSCL asks to minimize the total leasing costs. We assume the following configuration on the leases.

**Definition 1.** *(Lease Configuration) Leases of type $l$ only start at times $t$ with $t \equiv 0 \mod d_l$, where $d_l$ is the length of lease type $l$. Moreover, all lease lengths are power of two.*

This configuration has been similarly defined by Meyerson for the *Parking Permit* problem (Meyerson, 2005), who showed that by assuming this configuration, one loses only a constant factor in the competitive ratio. A similar argument can be easily made for OSCL, as was the case for all generalizations of the *Parking Permit* problem (Abshoff et al., 2016; Bienkowski et al., 2017; Nagarajan and Williamson, 2013).

**Preliminaries.** Our algorithm for OSCL is based on running Alon *et al.*'s (Alon et al., 2003) deterministic algorithm for the *Online Set Cover* problem (the weighted case), which constructs a fractional solution that is rounded online into an integral deterministic solution. Alon *et al.*'s algorithm has an $O(\log m \log n)$-competitive ratio and requires the knowledge of the set cover instance to make it deterministic. What is unknown to the algorithm is the order and subset of arriving elements. We will transform an instance $\alpha$ of OSCL into an instance $\alpha'$ of the *Online Set Cover* problem and run Alon *et al.*'s deterministic algorithm on $\alpha'$. An instance of the *Online Set Cover* problem consists of a universe of elements and a collection of subsets of the universe - an element of the universe arrives in each step. The algorithm needs to purchase subsets such that each arriving element is covered, upon its arrival, by one of these subsets, while minimizing the total costs of subsets. The algorithm may end up covering elements that never arrive.

**Algorithm.** Suppose the algorithm is given a universe $\mathcal{U}$ of elements and a collection $\mathcal{S}$ of subsets of $\mathcal{U}$. If there is one lease type, of infinite lease length ($\mathcal{L} = 1$), we have exactly an instance of the *Online Set Cover* problem and so Alon *et al.*'s (Alon et al., 2003) deterministic algorithm would solve it. Otherwise, we do the following - we represent each element $e \in \mathcal{U}$ by $n$ pairs, one for each of the at most $n$ potential time steps at which $e$ can arrive. We let pair $(e,t)$ represent element $e$ at time step $t$. We denote by $\mathcal{N}$ the collection of all these pairs. A subset $S \in \mathcal{S}$ can be leased using lease type $l$ for cost $c_l$ and remains *active* for $d_l$ time steps. We represent subset $S$ of lease type $l$ at time $t$ as a triplet $(S,l,t)$. We denote by $\mathcal{M}$ the collection of all these triplets. We now construct an instance of the *Online Set Cover* problem with $\mathcal{N}$ and $\mathcal{M}$ being the collection of elements and of subsets, respectively. Pair $(e,t)$ can be covered by triplet $(S,l,t')$ if $e \in S$ and $t \in [t', t' + d_l]$. When an element arrives at time $t$, pair $(e,t)$ is given as input to the *Online Set Cover* instance for step $t$. Note that each element $e \in \mathcal{U}$ arrives only once. An algorithm for the *Online Set Cover* problem will ensure that $e's$ corresponding pair at the time it arrives is covered. Moreover it will ignore (not necessarily cover) all other pairs corresponding to the other time steps and this is equivalent to having elements that never arrive in an *Online Set Cover* instance. Hence, running Alon *et al.*'s (Alon et al., 2003) algorithm will yield a feasible deterministic solution for OSCL.

**Competitive Analysis.** OSCL has a lower bound of $\Omega(\log m \log n + L)$ resulting from the $\Omega(\log m \log n)$ lower bound for OSC (Korman, 2005), where $m$ is the number of subsets and $n$ is the number of elements, and the $\Omega(L)$ lower bound for the *Parking Permit* problem (Meyerson, 2005), where $L$ is the number of lease types.

We fix any interval $I$ of length $\sigma$ and show that the algorithm would be $O(\log \sigma \log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive if this interval were the entire input, where $l_1$ is the length of the shortest lease, $\sigma$ is the length of the longest lease, $\mathcal{L}$ is the number of lease types, and $m$ is the number of subsets. Since all leases including the ones in the optimal solution end at the end of $I$ due to the lease configuration defined earlier, this would imply that the algorithm has an $O(\log \sigma \log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive ratio. Note that there are at most $\sigma$ elements over $I$, since at most one element arrives in each time step. The competitive ratio $O(\log|\mathcal{M}|\log|\mathcal{N}|)$ of the algorithm follows directly by setting the number of elements and subsets to $|\mathcal{N}|$ and $|\mathcal{M}|$, respectively. Now, we have that $|\mathcal{N}| = \sigma^2$ since there are $\sigma^2$ pairs in total. Next, we give an upper bound to $|\mathcal{M}|$ over $I$.

$$|\mathcal{M}| \leq m \cdot (\sum_{j=1}^{\mathcal{L}} \left\lceil \frac{\sigma}{l_j} \right\rceil)$$

Since $l_j$s are increasing and powers of two, we conclude that the sum above can be upper bounded by the sum of a geometric series with a ratio of $1/2$.

$$\sum_{j=1}^{\mathcal{L}} \left\lceil \frac{\sigma}{l_j} \right\rceil \leq \mathcal{L} + \sigma \left[ \frac{1}{l_1} \left( \frac{1-(1/2)^{\sigma}}{1-1/2} \right) \right] = \mathcal{L} + \sigma \left[ \frac{2}{l_1} \left( 1 - (1/2)^{\mathcal{L}} \right) \right]$$

Since $\mathcal{L} \geq 1$, we have:

$$\mathcal{L} + \sigma \left[ \frac{2}{l_1} \left( 1 - (1/2)^{\mathcal{L}} \right) \right] \leq \mathcal{L} + \frac{2\sigma}{l_1}.$$

Therefore, $|\mathcal{M}| \leq m \cdot (\mathcal{L} + \frac{2\sigma}{l_1})$, and the theorem below follows. □

**Theorem 3.** *There is a deterministic $O(\log\sigma \log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive algorithm for the Online Set Cover Leasing problem, where m is the number of subsets, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length.*

# 6 ONLINE CONNECTED DOMINATING SET LEASING (OCDSL)

**Definition.** Given a connected graph $G = (V,E)$, a sequence of disjoint subsets of $V$ arriving over time, and a set of $\mathcal{L}$ different lease types, each characterized by a duration and cost. A node can be leased using lease type $l$ for cost $c_l$ and remains active for $d_l$ time steps. A subset $S$ of nodes of $V$ serves as a connected dominating set of a given subset $D$ of $V$ if every node in $D$ is either in $S$ or has an adjacent node in $S$, and the subgraph induced by $S$ is connected in $G$. Each time step $t$, a subset of $V$ arrives and needs to be served by a connected dominating set of nodes active at time $t$. OCDSL asks to grow a connected dominating set with mininum leasing costs. OCDS is equivalent to OCDSL with one lease type ($L = 1$) of infinite length. Note that in both OCDS and OCDSL, the algorithm ends up purchasing (leasing) nodes that form one connected subgraph - the difference is that in OCDSL, at a certain time step $t$, only the currently active nodes needed to serve the nodes given at time $t$, are connected by nodes active at time $t$, to at least one of the previously leased nodes, thus maintaining one single connected subgraph.

We assume the lease configuration introduced earlier in Definition 1.

**Algorithm.** The algorithm assigns, at the first time step, any of the nodes leased by the algorithm, as a root node $s$. At time step $t$:

**Input:** $G = (V,E)$, subset $D_t$ of $V$
**Output:** A set of leased nodes that form a connected dominating set of $D_t$

1. Lease a set $DS_t$ of nodes that form a minimal dominating set of $D_t$.
2. Assign to each node in $DS_t$ a *connecting* node, that is any adjacent node from the set $D_t$. Buy the cheapest lease for each of these connecting nodes. If $t = 1$, assign any of the nodes in $DS_t$ as a root node $s$.
3. Lease a set of nodes that connect all connecting nodes to $s$.

**Algorithm Description.** To find a set of leased nodes that form a *minimal dominating set* of a subset $D_t$, we run our deterministic algorithm for *Online Set Cover Leasing* presented in Section 5. An *Online Set Cover Leasing* instance is formed by making each node an element, and corresponding each node to a set that contains the node itself, along with its adjacent nodes - sets are leased with $\mathcal{L}$ different lease types. Our algorithm for *Online Set Cover Leasing* has an $O(\log\sigma \log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive ratio, where $m$ is the number of subsets, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length.

To find a set of leased nodes that connect a subset of nodes to $s$, we run the deterministic algorithm for *Online Steiner Tree Leasing* problem (OSTL) by Bienkowski *et al.* (Bienkowski et al., 2017), defined as follows. Given a connected graph $G = (V,E)$, a root node $s$, a sequence of nodes of $V$ (called *terminals*) arriving over time, and a set of $\mathcal{L}$ different lease types, each characterized by a duration and cost. An edge can be leased using lease type $l$ for cost $c_l$ and remains active for $d_l$ time steps. Each step $t$, a node arrives and needs to be connected to $s$ through a path of edges active at time $t$. OSTL asks to minimize the total leasing costs. The algorithm by Bienkowski *et al.* (Bienkowski et al., 2017) has an $O(\mathcal{L}\log k)$-competitive ratio, where $k$ is the number of terminals.

The *Online Steiner Tree Leasing* problem studied by Bienkowski *et al.* (Bienkowski et al., 2017) is for edge-weighted graphs and the algorithmic cost is measured by adding the leasing costs of the edges and not the nodes. Our model in this paper assumes no weights on the nodes, and hence the competitive ratio given by Bienkowski *et al.* (Bienkowski et al., 2017) for edge-weighted graphs carries over to our graph model in this paper. This would not have been the case had there been non-uniform weights on the nodes since the node-weighted variant of the *Online Steiner Tree Leasing* problem generalizes the edge-weighted variant. Hence, whenever the algorithm for *Online Steiner Tree Leasing* leases an edge $(u,v)$ at time $t$ with lease type $l$, we lease both $u$ and $v$ at the same time $t$ with the same lease type $l$ and hence the cost will only double.

**Competitive Analysis.** Since OCDSL generalizes OSCL, $\Omega(\log^2 n + L)$ is a lower bound for OCDSL, where $n$ is the number of nodes and $L$ is the number of lease types. The proof of the competitive analysis is ommited due to lack of space.

**Theorem 4.** *There is a deterministic* $O\Big((\sigma+1)\cdot \log\sigma \log(n\mathcal{L} + 2n\frac{\sigma}{l_1}) + \mathcal{L}\cdot\log n\Big)$ *competitive algorithm for the Online Connected Dominating Set Leasing problem, where n is the number of nodes, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length.*

## 7 ONLINE r-hop CONNECTED DOMINATING SET LEASING (r-hop OCDSL)

**Definition.** Given a connected graph $G = (V,E)$, a positive integer $r$, a sequence of disjoint subsets of $V$ arriving over time, and a set of $\mathcal{L}$ different lease types, each characterized by a duration and cost. A node can be leased using lease type $l$ for cost $c_l$ and remains active for $d_l$ time steps. A subset $S$ of nodes of $V$ serves as an $r$-hop connected dominating set of a given subset $D$ of $V$ if for every node $v$ in $D$, there is a vertex $u$ in $S$ such that there are at most $r$ hops between $v$ and $u$ in $G$, and the subgraph induced by $S$ is connected in $G$. Each time step $t$, a subset of $V$ arrives and needs to be served by an $r$-hop connected dominating set of nodes active at time $t$. r-hop OCDSL asks to grow an $r$-hop connected dominating set with minimum leasing costs.

OCDSL is equivalent to r-hop OCDSL for $r = 1$. We assume the lease configuration introduced earlier in Definition 1.

**Algorithm.** The algorithm assigns, at the first time step, any of the nodes leased by the algorithm, as a root node $s$. At time step $t$:

**Input:** $G = (V,E)$ and subset $D_t$ of $V$
**Output:** A set of leased nodes that form $r$-hop connected dominating set of $D_t$

1. Lease a set $rDS_t$ of nodes that form a minimal $r$-hop dominating set of $D_t$. If $t = 1$, assign any of the nodes in $rDS_t$ as a root node $s$.
2. Lease a set of nodes that connect all nodes in $rDS_t$ to $s$.

**Algorithm Description.** To find a set of leased nodes that form a *minimal r-hop dominating set* of a subset $D_t$, we run our deterministic algorithm for *Online Set Cover Leasing* presented in Section 5. An *Online Set Cover Leasing* instance is formed by making each node an element, and corresponding each node to a set that contains the node itself, along with all nodes that are at most $r$ hops away from it - sets are leased with $\mathcal{L}$ different lease types. Our algorithm for *Online Set Cover Leasing* has an $O(\log\sigma\log(m\mathcal{L} + 2m\frac{\sigma}{l_1}))$-competitive ratio, where $m$ is the number of subsets, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length. To find a set of leased nodes that connect a subset of nodes to $s$, we run the deterministic $O(\mathcal{L}\log k)$-competitive algorithm for *Online Steiner Tree Leasing* problem (OSTL) by Bienkowski *et al.* (Bienkowski et al., 2017), defined earlier.

**Competitive Analysis.** The only lower bound for r-hop OCDSL is the one for OCDSL, $\Omega(\log^2 n + L)$, where $n$ is the number of nodes and $L$ is the number of lease types. The proof of the competitive analysis is ommited due to lack of space.

**Theorem 5.** *There is a deterministic* $O\Big(\mathcal{L}(1+\sigma(2r-1))\log\sigma\log(n\mathcal{L}+2n\frac{\sigma}{l_1})\log n\Big)$*-competitive algorithm for the Online r-hop Connected Dominating Set Leasing problem, where n is the number of nodes, $\mathcal{L}$ is the number of lease types, $\sigma$ is the longest lease length, and $l_1$ is the shortest lease length.*

## 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.

Barman, S., Pal, M., and Mondal, S. (2018). An optimal algorithm to find minimum k -hop dominating set of interval graphs. *Discrete Mathematics, Algorithms and Applications*, 11.

Berman, P. and Coulston, C. (1997). Online algorithms for steiner tree problems (extended abstract). pages 344–353.

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.

Boyar, J., Eidenbenz, S. J., Favrholdt, L. M., Kotrbcík, M., and Larsen, K. S. (2016). Online dominating set. In *15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016, June 22-24, 2016, Reykjavik, Iceland*, pages 21:1–21:15.

Buchbinder, N. and Naor, J. (2005). Online primal-dual algorithms for covering and packing problems. In Brodal, G. S. and Leonardi, S., editors, *Algorithms – ESA 2005*, pages 689–701, Berlin, Heidelberg. Springer Berlin Heidelberg.

Coelho, R. S., Moura, P. F. S., and Wakabayashi, Y. (2017). The k-hop connected dominating set problem: approximation and hardness. *Journal of Combinatorial Optimization*, 34(4):1060–1083.

Daliri Khomami, M. M., Rezvanian, A., Bagherpour, N., and Meybodi, M. R. (2018). Minimum positive influence dominating set and its application in influence maximization: a learning automata approach. *Applied Intelligence*, 48(3):570–593.

D'Andrea, R. (2012). Guest editorial: A revolution in the warehouse: A retrospective on Kiva systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4):638–639.

Du, D.-Z. and Wan, P.-J. (2013). *Connected dominating set. Theory and applications*.

Feige, U. (1998). A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652.

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.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Guha, S. and Khuller, S. (1998). Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387.

Halawi, B., Mourad, A., Otrok, H., and Damiani, E. (2018). Few are as good as many: An ontology-based tweet spam detection approach. *IEEE Access*, PP:1–1.

Hamann, H., Markarian, C., Meyer auf der Heide, F., and Wahby, M. (2018). Pick, pack, & survive: Charging robots in a modern warehouse based on online connected dominating sets. In *9th International Conference on Fun with Algorithms, FUN 2018, June 13-15, 2018, La Maddalena, Italy*, pages 22:1–22:13.

Haraty, R. A., Dimishkieh, M., and Masud, M. (2015). An enhanced *k*-means clustering algorithm for pattern discovery in healthcare data. *IJDSN*, 11:615740:1–615740:11.

Haraty, R. A., Mansour, N., Moukahal, L., and Khalil, I. (2016). Regression test cases prioritization using clustering and code change relevance. *International Journal of Software Engineering and Knowledge Engineering*, 26(5):733–768.

Korman, S. (2005). On the Use of Randomization in the Online Set Cover Problem. Master's thesis, Weizmann Institute of Science, Israel.

Lund, C. and Yannakakis, M. (1994). On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981.

Markarian, C. and Meyer auf der Heide, F. (2019). Online algorithms for leasing vertex cover and leasing non-metric facility location. pages 315–321.

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.

Wagner, C., Singer, P., Karimi, F., Pfeffer, J., and Strohmaier, M. (2017). Sampling from social networks with attributes. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1181–1190, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Yu, J., Wang, N., Wang, G., and Yu, D. (2013). Connected dominating sets in wireless ad hoc and sensor networks - A comprehensive survey. *Computer Communications*, 36(2):121–134.