Algorithmic Study of Online Multi-Facility Location Problems

Christine Markarian\textsuperscript{1} · Abdul-Nasser Kassar\textsuperscript{2} · Manal Yunis\textsuperscript{2}

Received: 16 October 2021 / Accepted: 5 May 2022 / Published online: 19 May 2022
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

Abstract

Facility location (FL) is a well-known optimization problem that asks to optimally place facilities so as to serve clients at various locations, requesting a facility service, with minimum possible costs. Many variants of FL have been known, appearing as sub-problems in many applications in computer science, management science, and operations research. Most FL models studied thus far assume that clients need to be served by connecting each to one facility. To overcome facility failures and provide a robust solution, we investigate in this paper FL problems that require each client to be connected to multiple facilities, represented by an additional input parameter. The aim of the algorithm is then to provide a robust service to all clients while minimizing the total connecting and facility purchasing costs. This is known as the Multi-Facility Location problem (MFL) and has been studied in the offline setting, in which the entire input sequence is given to the algorithm at once. In this paper, we study MFL in the online setting, in which client requests are not known in advance but are revealed to the algorithm over time. We refer to it as the online multi-facility location problem (OMFL) and study its metric and non-metric variants. We propose the first online algorithms for these variants and measure their performance using the standard notion of competitive analysis. The latter is a worst-case analysis that compares the cost of the online algorithm to that of the optimal offline algorithm that is assumed to know all demands in advance. We further study OMFL in the leasing setting, in which facilities are leased, rather than bought, for different durations and prices, and each arriving client needs to be connected to multiple facilities leased at the time of its arrival. The aim is to minimize the total connecting and facility leasing costs.

Keywords Robustness · Multi-facility location · Online algorithms · Competitive analysis · Randomized rounding · Leasing framework

Introduction

Facility location (FL) is a classical NP-hard optimization problem widely spread in the fields of computer science and operations research \cite{11, 26}. In its simplest form, we are given a set of facilities and a set of clients. Each facility has an opening cost and each client \( i \) has a connecting cost to each facility \( j \), which is the distance between \( i \) and \( j \). The goal is to open a subset of the facilities and connect the clients to open facilities so as to minimize the sum of the facility costs and the connecting costs. FL is known to have two versions, metric and non-metric. In the metric version, the distances are assumed to be symmetric and satisfy the triangle inequality.

In this paper, we focus on the algorithmic study of FL problems and consider an online setting in which clients are not known in advance but revealed to the algorithm over time. As soon as one arrives, it needs to be connected. Many real-world applications that contain FL...
as a sub-problem have this online nature. Therein, one is expected to react to present demands whenever they arrive, without knowing about future demands. Maintaining a given optimization goal becomes more challenging in the face of such an uncertainty. This encourages the study of online algorithms [15] for FL problems.

Applications. During the COVID-19 pandemic, healthcare providing communities were challenged to make fast wise decisions as a reaction to the dynamic evolution of the disease. Not knowing what the future brings, many of these communities were forced to build new facilities with minimum possible costs, to serve patients as soon as a possible in most convenient ways. At the heart of such optimization decisions are online FL problems, solved with online algorithms.

Competitive analysis. In this paper, we use the standard framework for measuring online algorithms, known as competitive analysis, in which demands and their arrival order are selected by an oblivious adversary that is unaware of the choices of the algorithm. 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.

The study of Facility location (FL) in the online setting was initiated by Meyerson [33], who introduced the metric online facility location problem (OFL), and proposed an $O(\log n)$-competitive randomized algorithm, where $n$ is the number of clients. Alon et al. [3] studied the non-metric version and proposed an $O(\log n \log m)$-competitive randomized algorithm, where $n$ is the number of clients and $m$ is the number of facilities. Many other variations were known for both metric and non-metric variants in the online setting [1, 4, 10, 16, 17, 32].

Other variants include [18], in which the algorithm can merge existing facilities with each other, and only the decision of assigning some demands to the same facility is irrevocable. Another variant by [14] allows the algorithm to adapt the position of the facilities for costs proportional to the distance by which the position is changed.

All of these works assume that clients need to be served with one facility each. In many real-world applications, a robust service, in which a client is served with more than one facility, is desirable [19, 36, 37]. Facilities and/or connections to facilities may be prone to failure and assigning clients to multiple facilities would provide a fault-tolerant solution. Such solutions are sought in many applications, such as providing replicated cash data in distributed networks [7].

Online Setting

Motivated by the above, we explore in this paper a generalization of online Facility Location [3, 17], in which we are additionally given a parameter $k$, which is the number of facilities required to serve a client. We refer to it as the Online Multi-Facility Location problem, defined as follows.

Definition 1 (Online Multi-facility Location) We are given a collection of $m$ facilities, $n$ clients, and a positive integer $k$. Each facility has an opening cost and each client has a connecting cost to each facility. Clients arrive over time. As soon as one arrives, it needs to be connected to at least $k$ open facilities. To open a facility, we pay its opening cost. To connect a client to a facility, we pay the corresponding connecting cost. The goal is to minimize the total opening and connecting costs.

Contribution 1. We address the metric and non-metric variants of Online Multi-Facility Location. We provide the first online algorithms for these problems analyzed under the competitive analysis framework.

Our results for these variants can be summarized as follows:

- We refer to the non-metric version as Online Non-metric Multi-Facility Location (ONMFL). We propose an online $O(\log(kn) \log m)$-competitive randomized algorithm for ONMFL, where:
  - $m$ is the number of facilities
  - $n$ is the number of clients
  - $k$ is the number of required connections

  The algorithm uses a randomized rounding approach that first constructs a fractional solution and then rounds it into an integral one. Its competitive analysis is based on first comparing the fractional solution constructed by the algorithm to the optimal offline solution and then measuring the fractional solution in terms of the integral one.

- We refer to the metric version as Online Metric Multi-Facility Location (OMMFL). We propose an online $O(\max\{f_{\min}, f_{\max}\} \cdot k \cdot \frac{\log n}{\log \log n})$-competitive deterministic algorithm for OMMFL, where:
  - $n$ is the number of clients
  - $k$ is the number of required connections
  - $c_{\max}$ and $c_{\min}$ are the maximum and minimum connecting costs, respectively
  - $f_{\min}$ and $f_{\max}$ are the maximum and minimum facility costs, respectively
The idea of the algorithm is to ensure first that each client is connected to one facility by running an algorithm for metric online facility location (OFL). Then, the \( k - 1 \) remaining connections are made by choosing the cheapest possible facilities so as not to worsen the competitive ratio by much. Our approach can be seen as a general framework that transforms any given online algorithm for metric OFL into an algorithm for OMMFL, by losing a bounded factor in the competitive ratio. Lower Bounds. For the non-metric version, there is an \( \Omega(\log m \log n) \) lower bound, under the assumption that NP \( \not\subseteq \mathrm{BPP} \), where \( m \) is the number of facilities and \( n \) is the number of clients, due to the lower bound given for Online Set Cover [23]. For the metric version, there is a lower bound of \( \Omega(\frac{\log n}{\log \log n}) \), where \( n \) is the number of clients, due to the lower bound given for metric OFL [17].

**Online Leasing Setting**

Over the past decades, many markets have seen changes in their resource management by shifting to leasing their resources rather than buying them [8, 21, 24]. While leasing has offered high flexibility to companies, their optimization decisions have become more complex. Companies were now expected not only to decide which facilities to choose but also when to lease a facility and for how long, since facilities could only be used as long as they were leased. To capture such scenarios, the theoretical leasing framework has been introduced in [34]. In this framework, there are a number of lease durations, respecting the economy of scale, such that longer leases cost more but are cheaper per unit time. The online nature of these problems, in which future information can’t be known entirely in advance, requires challenging decisions.

Consider, for example, a company that decides to lease a resource for a year and then ends up not using it as often as expected. Had it known how many clients will show up this year, it would have purchased shorter leases instead. Hence, the challenge is to make provably good leasing decisions in the face of the inaccuracy of future requests.

Motivated by the above, we study online multi-facility location in the leasing setting. In the leasing variant of Online Multi-Facility Location, each arriving client needs to be served with multiple facilities leased at the time of its arrival. The aim is to minimize the total connecting and facility leasing costs.

**Contribution 2.** In this paper, we define Online Multi-Facility Location in the online leasing setting and study its metric and non-metric variants. We propose the first online algorithms for these problems analyzed under the competitive analysis framework.

**Overview**

The rest of the paper is organized as follows. In the next section, we give an overview of works related to OMMFL, present an algorithm for ONMFL, and prove its competitive ratio. In the following section, we give an overview of works related to OMMFL, present an algorithm for OMMFL, and prove its competitive ratio. In the next section, we describe the leasing framework and give some preliminaries. In the next section, we propose an online algorithm for leasing-ONMFL and prove its competitive ratio. In the next section, we propose an online algorithm for leasing-OMMFL and prove its competitive ratio. We conclude in the last section with a summary of the results and some future work.

This work is an extension of our results that appeared in the conference paper [29]. In this paper, we have defined OMFL and generalized its algorithms, for both the metric and non-metric variants, in the leasing framework. The next sections present the results obtained for this setting.

**Online Non-metric Multi-facility Location (ONMFL)**

In this section, we start with an overview of works related to ONMFL and give some preliminaries. Then we present an online randomized algorithm for ONMFL and analyze its competitive ratio.

**Preliminaries and Related Work**

OMMFL is a generalization of the Online non-metric facility location (ONFL) [3] with \( k = 1 \). Alon et al. [3] gave an \( O(\log n \log m) \)-competitive randomized algorithm for ONFL, where \( n \) is the number of clients and \( m \) is the number of facilities.

A closely related problem is the Online Set \( k \)-Multicover (OSMC) [5]. Given a universe \( U \) of \( n \) elements, a family \( S \) of \( m \) subsets of \( V \), each associated with a cost, and a positive integer \( k \). A subset \( D \subseteq U \) of elements arrives over time. OSMC asks to select a collection \( C \subseteq S \) of subsets, of minimum cost, such that each arriving element belongs to at least \( k \) subsets of \( C \). Berman and DasGupta [5] proposed an \( O(\log m \log d) \)-competitive randomized algorithm for OSMC, where \( m \) is the number of subsets and \( d \) is the maximum set size.

Transformations between OSMC and ONMFL instances can be made in both directions. An instance of ONMFL can be transformed into an instance of OSMC as follows. We associate each facility with each of the \( 2^m - 1 \) possible groups of clients, and let each facility/group be a subset, with cost equal to the sum of the cost of the facility and the connecting costs of the clients in the group to the facility.
We let each client be an element. Following this transformation, the algorithm of Berman and DasGubta would imply a feasible algorithm for ONMFL, but with competitive ratio \( O(\log(m(2^n - 1)) \log n) \), where \( n \) is the number of clients, and \( m \) is the number of facilities. An instance of OSMC can be transformed into an instance of ONMFL as follows. We represent each subset by a facility and let the opening cost be the subset cost. We represent each element by a client and set the connecting cost to 0 if the client belongs to the subset/facility and infinity otherwise. The offline version of OSMC, in which all elements are given in advance, has an \( O(\log d) \) approximation [22, 38], where \( d \) is the maximum set size. Similar transformations can be made in the offline setting. This implies an \( O(\log n) \) approximation for the offline version of ONMFL, where \( n \) is the number of clients. These are the best achievable unless \( \text{NP} \subset \text{DTIME}(n^{\log \log n}) \) [12].

The Online Set Cover (OSC) [2] is a special case of OSMC with \( k = 1 \). Korman’s lower bound on the competitive ratio of OSC [23] implies an \( \Omega(\log m \log n) \) lower bound on the competitive ratio of ONMFL, under the assumption that \( \text{NP} \not\subset \text{BPP} \), where \( m \) is the number of facilities and \( n \) is the number of clients. For other online problems solved with similar techniques as the ones used in this paper, see [20, 30].

### Online Algorithm

Our algorithm for ONMFL is based on constructing a fractional solution first and then rounding it online into an integral solution. Unlike [2] and [3] for Online Set Cover (OSC) and Online Non-metric Facility Location (ONFL), respectively, in which a potential function is used in the competitive analysis of the algorithms, our analysis is based on simple arguments, similar to those we gave in our previous work for a related problem, the Online Node-weighted Steiner Forest [28].

We start by giving the following graph formulation for ONMFL. We refer the reader to Fig. 1 for an example graph. There is a root node \( r \), \( m \) facility nodes, and \( n \) client nodes. We add a directed edge from \( r \) to each facility node \( j \), of cost equal to the opening cost of facility \( j \). We add a directed edge from each facility node \( j \) to each client node \( i \), of cost equal to the connecting cost of client \( i \) to facility \( j \). As soon as a client \( i \) arrives, we need to purchase \( k \) disjoint paths from \( r \) to \( i \). The goal is to minimize the total costs of the paths purchased, where the cost of a path is the cost of its edges. To output a solution for ONMFL, each facility whose corresponding edge is purchased will be opened, and each client whose corresponding edge to an open facility is purchased will be connected to that facility.

The algorithm initially knows \( n \), the number of clients; the number of required connections \( k \); and the opening costs of facilities. A subset \( D \) of \( n' \leq n \) clients arrives over time.

![Graph formulation of online non-metric multi-facility location](image)

As soon as a client \( i \) arrives, the algorithm is given the connecting costs of \( i \) to each facility, and is expected to react.

Let \( G = (V, E) \) be the graph constructed. Each edge added to \( E \) is given a fraction, set initially to 0. The algorithm does not allow these fractions to decrease over time. These form a fractional solution for ONMFL. The maximum flow from node \( u \) to node \( v \) in \( G \) is the smallest total fraction of edges which if removed would disconnect \( u \) from \( v \). These edges form a minimum cut from \( u \) to \( v \) in \( G \). Let \( c_e \) and \( f_e \) be the cost and fraction of edge \( e \), respectively. A path is purchased if and only if its edges are purchased.

Random process. The algorithm makes its random choices, based on \( a \), the minimum among \( 2[\log(kn + 1)] \) independently chosen random variables, distributed uniformly in the interval \([0, 1]\). The choice of \( 2[\log(kn + 1)] \) will be made clear in the competitive analysis later. Intuitively, we make the analysis for a single client first and then extend it to the at most \( n \) clients, each connected to \( k \) facilities.

Next, we describe how the algorithm reacts upon the arrival of a new client. The equation below in the algorithm shows how the fractions of the edges in the constructed minimum cut are increased based on the cost of each edge and the total cardinality of the corresponding minimum cut.

**Input:** \( G = (V, E) \) and client node \( i \in D \)

**Output:** Set of edges purchased

Make a copy \( G' \) of \( G \): As long as there are \(< k \) disjoint paths purchased from \( r \) to \( i \) in \( G \), do the following:

1. While the maximum flow from \( r \) to \( i \) in \( G' \) is less than 1, construct a minimum cut \( Q \) from \( r \) to \( i \) in \( G' \); for each edge \( e \in Q \), make the following fraction increase:
   \[
   f_e = f_e \cdot (1 + c_e) + \frac{1}{|Q| \cdot c_e}
   \]
2. Purchase each edge \( e \) with \( f_e > a \).
3. If there is no purchased path from \( r \) to \( i \) in \( G' \), find a minimum-cost such path and purchase it.
4. Refer to all facilities whose corresponding edges were purchased as *open*; delete from \( G' \) the purchased edges from each open facility to \( i \).

**Competitive Analysis**

The algorithm buys edges in the second and third steps. In the second step, its choices are made based on the random process, whereas in the third step, its choices are made to guarantee a feasible solution. We measure the expected cost of each separately. Let \( Opt \) be the cost of the optimal offline solution and let \( frac \) be the cost of the fractional solution constructed by the algorithm in the first step.

**Choices based on random process:** Let \( S' \) be the set of edges purchased in the second step of the algorithm and let \( C_{S'} \) be its expected cost. These edges are purchased by the algorithm based on the random process described earlier. Let us fix an \( l : 1 \leq l \leq 2 \lfloor \log(\text{kn} + 1) \rfloor \) and an edge \( e \). We denote by \( X_{e,l} \) the indicator variable of the event that \( e \) is chosen by the algorithm based on the random choice of \( l \).

\[
C_{S'} = \sum_{e \in S'} \sum_{l=1}^{2 \lfloor \log(\text{kn} + 1) \rfloor} c_e \cdot \text{Exp}[X_{e,l}]
\]

\[
= 2 \lfloor \log(\text{kn} + 1) \rfloor \sum_{e \in S'} c_{e} f_{e}
\]

Notice that \( \sum_{e \in S'} c_{e} f_{e} \) is upper bounded by the cost of the fractional solution. The latter can be measured against the optimal offline solution, as follows. The idea here is that every time the algorithm performs a fraction increase, it does not exceed 2. Moreover, the total number of fraction increases can be measured in terms of the cost of the optimal offline solution.

The fraction increase contributed by each edge \( e \) in a minimum cut \( Q \) is \( \left( \frac{f_e}{c_e} + \frac{1}{|Q| c_e} \right) \). The algorithm would make a fraction increase only if the maximum flow is less than 1. This means we have that \( \sum_{e \in Q} f_{e} < 1 \) before a fraction increase. Therefore, each fraction increase does not exceed:

\[
\sum_{e \in Q} c_e \cdot \left( \frac{f_e}{c_e} + \frac{1}{|Q| c_e} \right) < 2
\]

As long as the algorithm hasn’t purchased at least \( k \) disjoint paths from \( r \) to a given client \( i \), it enters the loop that starts by constructing a maximum flow on the graph \( G' \). Notice how \( G' \) shrinks over time, as the algorithm purchases the paths.

**Lemma 1** Whenever the algorithm makes a fraction increase, \( G' \) contains at least one path that is also in the optimal offline solution.

**Proof** To see why this holds, fix any time \( t \) before a fraction increase. Let \( s < k \) be the number of disjoint paths purchased by the algorithm at time \( t \). \( G' \) at time \( t \) must contain at least one optimal path since \( G' \) is constructed by removing \( s \) (less than \( k \)) feasible paths from \( G \) and the optimal offline solution contains at least \( k \) disjoint paths in \( G \).

Finally, the algorithm would have an edge \( e \) from the optimal offline solution in every minimum cut \( Q \) of \( G' \), since \( Q \) must contain an edge from each path, by definition. Based on the equation for the fraction increase, after \( O(\log |Q|) \) fraction increases, the fraction \( f_e \) of \( e \) becomes 1, and no further increases can be made, as \( e \) will not be in any future minimum cut. The size of any minimum cut is upper bounded by \( m \), the number of facilities or the maximum available paths from \( r \) to client \( i \). We can now bound the fractional solution:

\[
frac{\text{frac}}{\text{Opt}} \leq O(\log m \cdot \text{Opt})
\]

Combining Eqs. 1, 2, and 3 imply an upper bound on the expected cost \( C_{S'} \) of the edges bought in the second step of the algorithm:

\[
C_{S'} \leq O(\log(\text{kn}) \log m \cdot \text{Opt})
\]

**Choices to guarantee feasible solution:** Let \( S'' \) be the set of edges purchased in the third step of the algorithm and let \( C_{S''} \) be its expected cost. These edges are purchased by the algorithm only if a path has not been bought by the random process in the second step. Every time the algorithm purchases a path in this step, its cost does not exceed \( Opt \) since the algorithm buys a minimum-cost path in \( G' \), and as we showed earlier in Lemma 1, \( G' \) contains at least one path that is also in the optimal solution.

- (one client, one path) We start by calculating the expected cost incurred by a single client for purchasing a single path. Fix a client \( i \). Let \( Q, j, 1 \) be a minimum cut of \( G' \) constructed after the algorithm has purchased \( j < k \) disjoint paths from \( r \) to \( i \) and has completed the first step. The probability of purchasing the \((j + 1)\)th path for a single \( 1 \leq l \leq 2 \lfloor \log(\text{kn} + 1) \rfloor \) is:

\[
\prod_{e \in Q, j+1} (1 - f_e) \leq e^{-\sum_{e \in Q, j+1} f_e} \leq 1/e
\]

Notice that the last inequality holds because the algorithm ensures that \( \sum_{e \in Q, j+1} f_e \geq 1 \) at the end of the first step (Max-flow min-cut theorem). The expected cost of purchasing the \((j + 1)\)th path for all \( 1 \leq l \leq 2 \lfloor \log(\text{kn} + 1) \rfloor \), is less than \( 1/(\text{kn})^2 \cdot \text{Opt} \).

- (one client, \( k \) paths) The expected cost of purchasing all \( k \) paths is the sum of the expected costs for each path and is less than \( k \cdot 1/(\text{kn})^2 \cdot \text{Opt} \).
– (total cost of all clients) The total expected cost incurred by all \( n' \) clients that arrived is less than:

\[
n' \cdot k \cdot \frac{1}{(kn)^2} \cdot \text{Opt} \leq n \cdot k \cdot \frac{1}{(kn)^2} \cdot \text{Opt} = \frac{1}{kn} \cdot \text{Opt}
\]

Therefore, the expected cost \( C_{\text{new}} \) of the edges bought in the third step of the algorithm is:

\[
C_{\text{new}} \leq \frac{1}{kn} \cdot \text{Opt}
\] (5)

Equations 4 and 5 yield to the following theorem.

**Theorem 1** There is an online \( \mathcal{O} (\log (kn) \log m) \)-competitive randomized algorithm for the Online Non-metric Multi-Facility Location, where \( m \) is the number of facilities; \( n \) is the number of clients; and \( k \) is the number of required connections.

### Online Metric Multi-facility Location (OMMFL)

In this section, we start with an overview of works related to OMMFL and give some preliminaries. Then we present an online randomized algorithm for OMMFL and analyze its competitive ratio.

While this problem has been intensively studied in the offline setting [7, 39], we are not aware of any online algorithm for it.

### Preliminaries and Related Work

OMMFL is a generalization of metric Online Facility Location (OFL) [16, 17, 33] with \( k = 1 \). Meyerson [33] introduced metric OFL and proposed an \( \mathcal{O} (\log n) \)-competitive randomized algorithm, where \( n \) is the number of clients. Fotakis [17] showed that no randomized online algorithm can achieve a competitive ratio better than \( \Omega (\frac{\log n}{\log \log n}) \) against an oblivious adversary and gave a deterministic algorithm with asymptotically matching \( \mathcal{O} (\frac{\log n}{\log \log n}) \)-competitive ratio. In another work later, he proposed a primal-dual deterministic algorithm with \( \mathcal{O} (\log n) \)-competitive ratio, that was simpler to formulate, analyze, and implement [16]. The competitive ratio we achieve for OMMFL is based on running the deterministic algorithm of Fotakis [17] for metric OFL.

The lower bound on the competitive ratio of metric OFL by Fotakis [17] implies an \( \Omega (\frac{\log n}{\log \log n}) \) lower bound on the competitive ratio of OMMFL.

### Online Algorithm

Let \( A_{\text{OFL}} \) be any online (deterministic or randomized) algorithm for metric Online Facility Location (OFL), with competitive ratio \( r \). Given an instance \( I \) of OMMFL with positive integer \( k \). Client \( i \) arrives. Our algorithm needs to connect \( i \) to \( k \) different open facilities.

1. The algorithm starts by running \( A_{\text{OFL}} \) on instance \( I \), where \( k = 1 \). This results in opening some facilities and connecting \( i \) to one open facility.
2. If \( i \) is the first client, we open the cheapest \( k - 1 \) facilities other than the one \( i \) is connected to. From this point on, there are at least \( k \) open facilities. According to \( A_{\text{OFL}} \), these facilities remain closed unless opened by \( A_{\text{OFL}} \).
3. The algorithm will then connect \( i \) to any other \( k - 1 \) open facilities.

### Competitive Analysis

Let \( I \) be an instance of OMMFL with positive integer \( k \). Let \( I' \) be the same instance as \( I \) except for \( k = 1 \). Let \( \text{Opt} \) and \( \text{Opt}' \) be the cost of the optimal solution for \( I \) and that for \( I' \), respectively. Let \( C \) and \( C' \) be the cost of our algorithm for \( I \) and that of \( A_{\text{OFL}} \) for \( I' \), respectively. We denote by \( C_{\text{fac}} \) and \( C_{\text{con}} \) the costs incurred by our algorithm to open facilities and to connect clients, respectively. We denote by \( C'_{\text{fac}} \) and \( C'_{\text{con}} \) the costs incurred by \( A_{\text{OFL}} \) to open facilities and to connect clients, respectively.

The algorithm opens the cheapest \( k - 1 \) facilities other than the ones opened by \( A_{\text{OFL}} \). Let \( f_{\text{max}} \) be the maximum facility cost and \( f_{\text{min}} \) the minimum facility cost. Thus, we have that:

\[
C_{\text{fac}} \leq C'_{\text{fac}} + f_{\text{max}} \cdot (k - 1)
\] (6)

Given any client \( i \). Apart from its connecting cost \( c_i \) incurred by \( A_{\text{OFL}} \), our algorithm connects \( i \) to \( k - 1 \) other facilities, each resulting in a connecting cost at most \( \frac{c_{\text{max}}}{c_{\text{max}}} \cdot c_i \), where \( c_{\text{max}} \) and \( c_{\text{min}} \) are the maximum and minimum connecting costs, respectively. This implies an overall connecting cost:

\[
C_{\text{con}} \leq C'_{\text{con}} \cdot \left(1 + \frac{c_{\text{max}}}{c_{\text{min}}} \cdot (k - 1) \right)
\] (7)

We now add the two equations above and do some algebraic manipulations using:

\[
\begin{align*}
C_{\text{con}}' &\leq C' \leq C'_{\text{con}} \\
C_{\text{fac}}' &\leq C'_{\text{fac}} \leq C' \geq f_{\text{min}}
\end{align*}
\]

This yields to:
Given an online (deterministic or randomized) r-competitive algorithm for metric Online Facility Location. There is an online
\[ O(\max\{\frac{f_{\max}}{f_{\min}}, \frac{c_{\max}}{c_{\min}}\} \cdot k \cdot r) \]-competitive algorithm for the Online Metric Multi-Facility Location, where k is the number of required connections; c_{\max} and c_{\min} are the maximum and minimum connecting costs, respectively; and f_{\max} and f_{\min} are the maximum and minimum facility costs, respectively.

Running the deterministic algorithm of Fotakis \[17\] for metric OFL, with \( O(\frac{\log n}{\log \log n}) \)-competitive ratio, results in the following.

\textbf{Corollary 1} There is an online \( O(\max\{\frac{f_{\max}}{f_{\min}}, \frac{c_{\max}}{c_{\min}}\} \cdot k \cdot \frac{\log n}{\log \log n}) \)-competitive deterministic algorithm for the Online Metric Multi-Facility Location, where n is the number of clients; k is the number of required connections; c_{\max} and c_{\min} are the maximum and minimum connecting costs, respectively; f_{\max} and f_{\min} are the maximum and minimum facility costs, respectively.

\section*{The Leasing Framework}

We dedicate the rest of the paper to studying the Online Multi-Facility Location problem in the leasing setting. The leasing framework was first introduced by Meyerson \[34\] with a simple leasing problem, known as the Parking Permit problem. In this framework, rather than purchasing resources (facilities in our case), resources/facilities are leased for different durations and prices, such that as soon as a lease for some facility expires, the facility is not anymore able to serve any clients unless a new lease is purchased for the facility. That is, a facility can only be used on days at which the algorithm has purchased a lease for it.

In this section, we describe the leasing framework by providing some preliminaries and related works. Then, we introduce the leasing variant of Online Multi-Facility Location. We refer to its non-metric and metric leasing variants as Online Leasing Non-metric Multi-Facility Location (leasing-ONMFL) and Online Leasing Metric Multi-Facility Location (leasing-OMMFL), respectively. We propose the first online algorithms for these problems and analyze their competitive ratio. These algorithms are designed by extending the algorithms presented in Sections “Online Non-metric Multi-facility Location (ONMFL)” and “Online Metric Multi-facility Location (OMMFL)” for ONMFL and OMMFL, respectively.

\section*{Preliminaries and Related Work}

In the leasing framework, a resource or facility is leased for use at different durations and prices rather than being purchased. There are L lease types, each characterized by a duration and cost, respecting the economy of scale, such that a longer lease costs more but less per unit time. That is, if it is known in advance that a resource will be regularly needed for a duration of the longest lease type (e.g., 1 year), it would be cheaper if it is purchased to cover the demands of that year rather than purchasing short-duration leases (e.g., twelve 1-month leases or four 3-month leases). Due to the unpredictable nature of real-world leasing scenarios, no information about future demands is known in advance, and this imposes the main challenge in making the leasing decisions.

\subsection*{Lower Bounds.} The leasing framework was initiated by Meyerson \[34\], who proposed deterministic and randomized online algorithms for the Parking Permit problem (PP), and proved matching lower bounds, in terms of the number of lease types L. In particular, Meyerson proved lower bounds of \( \Omega(L) \) and \( \Omega(\log L) \) on the competitive ratio of any deterministic and any randomized algorithm for PP, respectively. Leasing-ONMFL and leasing-OMMFL generalize PP and hence the lower bounds on the competitive ratio of PP algorithms also apply to the algorithms for leasing-ONMFL and leasing-OMMFL.

\subsection*{Interval Model.} Meyerson \[34\] showed that, given any PP instance, it is possible to assume a special structure on the lease types, by losing only a constant factor in the competitive ratio. This has simplified the problem and facilitated the competitive analysis of the online algorithms. Consequently, this structure has been assumed in all leasing optimization problems built upon the leasing framework of Meyerson \[1, 6, 30–32, 35\].

The latter was known as the Interval Model. In this model, leases of the same type do not overlap and all have length power of two. In this paper, we also assume that lease types are in accordance with the Interval Model of Meyerson \[34\].

\subsection*{Related Work} Following Meyerson’s work, many online optimization problems were studied in the leasing framework, including Online Connected Facility Location Problem \[35\], Online Set Cover \[1, 30\], Online Vertex Cover \[32\], Online Non-metric Facility Location \[31, 32\], Online Metric
Facility Location [1], Online Steiner Tree [6], and Online Connected Dominating Set [30].

There have also been extensions to the original leasing framework. In [9], arbitrary demands are given and require multiple leases (permits) to be served. In [25], demands have deadlines and need to be served any time before their deadline. In [13], lease prices change over time. In [27], two models were studied, one in which not all demands need to be served and another in which demands can be delayed to be served against penalties. Other leasing models have been introduced in the context of facility services inspired by the COVID-19 pandemic as in [31].

We are now ready to define the leasing variant of Online Multi-Facility Location problem, or leasing-OMFL.

**Definition 2** (Leasing Variant of Online Multi-Facility Location, or leasing-OMFL) We are given a collection of \( m \) facilities, \( n \) clients, a positive integer \( k \), and \( L \) lease types, each characterized by lease length and lease price. Leasing a facility with lease type \( i \) incurs a lease price of \( p_i \) and can be used for a duration of lease length \( l_i \). Clients arrive over time. As soon as one arrives, it needs to be served by connecting it to at least \( k \) different facilities leased at the time of the client’s arrival. In each step, the algorithm is to decide which facilities to lease and for how long such that all clients are served. Each client has a connecting cost to each facility. To connect a client to a facility, we pay the corresponding connecting cost. The goal is to minimize the total leasing and connecting costs.

Note that leasing-OMFL extends OMFL to cover durations to infinity in the case of single lease types. Hence, the lower bounds associated with OMFL carry over to leasing-OMFL.

**Online Leasing Non-metric Multi-facility Location (Leasing-ONMFL)**

In this section, we first present an online algorithm for Online Leasing Non-metric Multi-Facility Location (leasing-ONMFL), and then analyze its competitive ratio.

The algorithm is based on the algorithm presented in ‘Overview’ for ONMFL and so we only describe the parts that are different. The graph formulation of leasing-ONMFL is as follows. We refer the reader to Fig. 2 for an example graph.

The nodes of the graph are:

- Root node \( r \)
- Two facility nodes for each facility, the first called actual facility node and the second called virtual facility node (a total of \( 2m \) facility nodes).

The edges of the graph are:

- We add a directed edge from \( r \) to each actual facility node, of cost 0.
- We add \( L \) directed edges from each actual facility node to its corresponding virtual facility node, representing each of the \( L \) lease types. The cost of each edge is equal to the corresponding lease type cost. Note that the idea behind having two nodes representing a facility is to be able to represent in a simple way the facility lease costs in a form of edge weights.
- We add a directed edge from each virtual facility node to each client node, of cost equal to the corresponding connecting cost of the client to the facility.

Upon the arrival of a new client, the algorithm needs to purchase \( k \) disjoint paths from \( r \) to the corresponding client node. Moreover, regarding the edges corresponding to the \( L \) lease types, the algorithm disregards all lease intervals that do not intersect with the client’s arrival time. Let \( G = (V, E) \) be the graph constructed. The introductory parts of the algorithm are similar to those in ‘Overview’ and so we do not mention them here again. As for the algorithm below, minor changes are made in comparison to the algorithm in ‘Overview’.

**Input:** \( G = (V, E) \) and client node \( i \in D \)

**Output:** Set of edges purchased

Make a copy \( G' \) of \( G \): As long as there are \( < k \) disjoint paths purchased from \( r \) to \( i \) in \( G \), do the following:

1. While the maximum flow from \( r \) to \( i \) in \( G' \) is less than 1, construct a minimum cut \( Q \) from \( r \) to \( i \) in \( G' \); for each edge \( e \in Q \), make the following fraction increase:
\[ f_e = f_e \cdot (1 + 1/c_e) + \frac{1}{|Q| \cdot c_e} \]

2. Purchase each edge \( e \) with \( f_e > \alpha \). Recall that \( \alpha \) is the random number generated before any client arrives.
3. If there is no purchased path from \( r \) to \( i \) in \( G' \), find a minimum-cost such path and purchase it.
4. There is at least one purchased path at this point. Delete from \( G' \) the actual and virtual facility nodes that belong to this path. Delete all \( L \) edges in between the two.

The competitive analysis of the algorithm differs slightly from the analysis of the algorithm in ‘Overview’. The difference is in the size of any minimum cut constructed. The latter is upper bounded by \( m \cdot L \) instead of \( m \), due to the \( L \) edges added between the actual and virtual facility nodes. This implies the following bound on the fractional solution. Notice that the \( m \) in Eq. 3 is now replaced by \( mL \).

\[
frac{\text{frac}}{\text{Opt}} \leq O(\log(mL))
\]

Hence, we conclude the following theorem.

**Theorem 3** There is an online \( O(\log(kn) \log(mL)) \)-competitive randomized algorithm for Online Leasing Non-metric Multi-Facility Location problem (leasing-ONMFL), where \( k \) is the number of required connections; \( n \) is the number of clients; \( m \) is the number of facilities; and \( L \) is the number of lease types.

**Online Leasing Metric Multi-facility Location (Leasing-OMMFL)**

In this section, we present an online algorithm for Online Leasing Metric Multi-Facility Location (leasing-OMMFL) and analyze its competitive ratio.

Let \( A_{MOFL} \) be any online (deterministic or randomized) algorithm for Metric Online Facility Leasing (MOFL), with competitive ratio \( r \). MOFL is a special case of leasing-OMMFL where \( k = 1 \) and has been studied in [1].

Given an instance \( I \) of leasing-OMMFL with positive integer \( k \). Client \( i \) arrives. Our algorithm needs to connect \( i \) to \( k \) different leased facilities.

1. The algorithm starts by running \( A_{MOFL} \) on instance \( I \), where \( k = 1 \). This results in buying leases for some facilities and connecting \( i \) to one leased facility.
2. Let \( j \) be the facility \( i \) is connected to, leased with lease type \( s \). The algorithm leases \( k - 1 \) other facilities other than the one \( i \) is connected to, with lease type \( s \). According to \( A_{MOFL} \), these leases remain unpurchased unless purchased by \( A_{MOFL} \).
3. The algorithm connects \( i \) to the \( k - 1 \) other leased facilities.

We use the same notations as those used in ’Online Non-metric Multi-facility Location (ONMFL)’, replacing opening costs by leasing costs. We show the parts that are different. Since the algorithm buys the same lease type for the \( k - 1 \) other facilities, the cost of leasing facilities would be bounded as follows.

\[
C_{\text{fac}} \leq C_{\text{fac}}' + (k - 1) \cdot C_{\text{fac}}'
\]

The analysis for the connecting costs would be the same as in ’Online Non-metric Multi-facility Location (ONMFL)’. Hence, using similar algebraic manipulations as before, we can conclude the theorem below.

**Theorem 4** Given an online (deterministic or randomized) \( r \)-competitive algorithm for Metric Online Facility Leasing. There is an online \( O(\frac{c_{\max}}{c_{\min}} \cdot k \cdot r) \)-competitive algorithm for Online Leasing Metric Multi-Facility Location (leasing-OMMFL), where \( c_{\max} \) and \( c_{\min} \) are the maximum and minimum connecting costs, respectively; and \( k \) is the number of required connections.

Running the deterministic algorithm for Metric Online Facility Leasing (MOFL) in [1], which has an \( O(l_{\max} \log l_{\max}) \)-competitive ratio, results in the corollary below.

**Corollary 2** There is an online \( O(\frac{c_{\max}}{c_{\min}} \cdot k \cdot l_{\max} \log l_{\max}) \)-competitive deterministic algorithm for Online Leasing Metric Multi-Facility Location (leasing-OMMFL), where \( c_{\max} \) and \( c_{\min} \) are the maximum and minimum connecting costs, respectively; \( k \) is the number of required connections; and \( l_{\max} \) is the longest lease length.

**Concluding Thoughts**

Tables 1 and 2 give a summary of the results obtained for the online setting and the online leasing setting, respectively.

In this paper, we have assumed there is a unique positive integer \( k \) for all clients. In many application scenarios, it is likely that clients have different number of required connections. A slight modification in our algorithms would yield to \( O(\log(k_{\max} n) \log m) \) and \( O(\max\{\frac{l_{\max}}{l_{\min}}, \frac{k_{\max}}{k_{\min}} \cdot \log \frac{\log n}{\log \log n}\}) \) competitive ratios for ONMFL and OMMFL, respectively, where \( k_{\max} \) is the maximum required connections. One research direction is to target better competitive ratios for these variants.
This brings us to the next question, for Online Metric Multi-Facility Location (OMMFL), whether it is possible to get rid of the parameters $c_{\max}$, $c_{\min}$, $f_{\max}$, and $f_{\min}$ from the competitive ratio or achieve lower bounds in terms of these parameters. To achieve the former, one may want to attempt a primal-dual approach, for instance, by trying to extend the algorithm of Fotakis [16] for metric Online Facility Location. Similarly, for leasing-OMMFL, one may want to attempt achieving a competitive ratio that is independent of $c_{\max}$ and $c_{\min}$.

Furthermore, in our model here, demands and their arrival order are given by an oblivious adversary. One may want to consider other types of adversary for these sequences, for instance, by exploring various probability distributions.

We have initiated in this paper the study of Online Multi-Facility Location in the online leasing framework. As mentioned earlier, the leasing framework has been investigated as a number of models, such as demands with deadlines [25], lease prices changing over time [13], and demands requiring multiple leases [9]. One may want to extend the study of Online Multi-Facility Location to these models.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**References**

1. Abshoff S, Kling P, Markarian C, Meyer auf der Heide F, Pietrzyk P. Towards the price of leasing online. J Comb Optim. 2016;32(4):1197–216.

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

3. Alon N, Awerbuch B, Azar Y, Buchbinder N, Naor JS. A general approach to online network optimization problems. ACM Trans Algorithms. 2006;2(4):640–60.

4. Anagnostopoulos A, Bent R, Upfal E, Van Hentenryck P. A simple and deterministic competitive algorithm for online facility location. Inf Comput. 2004;194(2):175–202.

5. Berman P, DasGupta B. Approximating the online set multi-cover problem via randomized winnowing. Theor Comput Sci. 2008;393(1):54–71.

6. Bienkowski M, Kraska A, Schmidt P. A deterministic algorithm for online steiner tree leasing. In: Ellen F, Kolokolova A, Sack J-R, editors. Algorithms and Data Structures. pp. Cham: Springer International Publishing; 2017. p. 169–80.

7. Byrka J, Srinivasan A, Swamy C. Fault-tolerant facility location: A randomized dependent lp-rounding algorithm. In Eisenbrand F, Shepherd FB, editors, Integer Programming and Combinatorial Optimization, pp 244–257, 2010. Springer, Berlin.

8. Cotei C, Farhat J. The leasing decisions of startup firms. Rev Pac Basin Financ Mark Policies. 2017;20(04):1750022.

9. de Lima MS, Felice MCS, Lee O. On generalizations of the parking permit problem and network leasing problems. Electron Notes Discrete Math. 2017;62:225–30.

10. Divík I, Imreh C. Online facility location with facility movements. CEJOR. 2011;19(2):191–200.

11. Drezner Z. Facility Location - A Survey of Applications and Methods. Springer Series in Operations Research and Financial Engineering, 1995.

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

13. Feldkord B, Markarian C, Meyer auf der Heide F. Price fluctuation in online leasing. In International Conference on Combinatorial Optimization and Applications, pp 17–31. Springer, 2017.

14. Feldkord B, Meyer auf der Heide F. Online facility location with mobile facilities. In Proceedings of the 30th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA ’18, pp 373-381, New York, NY, USA, 2018. Association for Computing Machinery.

15. Fiat A, Woeginger GJ. Online algorithms: The state of the art. 1998:1442. Springer.

16. Fotakis D. A primal-dual algorithm for online non-uniform facility location. J Discrete Algorithms. 2007;5(1):141–8.

17. Fotakis D. On the competitive ratio for online facility location. Algorithms. 2008;50(1):1–57.

18. Fotakis D. Online and incremental algorithms for facility location. SIGACT News. 2011;42(1):97–131.

19. Gerodimos A. The Journal of the Operational Research Society. 1998;49(12):1303–4.

20. Hamann H, Markarian C, Meyer auf der Heide F, Wabbi M. Pick, pack, & survive: Charging robots in a modern warehouse based on online connected dominating sets. In Ito H, Leonardi S, Pagli L, Prencipe G, editors, 9th International Conference on Fun with Algorithms, FUN 2018, June 13-15, 2018, La Maddalena, Italy, volume 100 of LIPIcs, pp 22:1–22:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

21. Hendel I, Lizzeri A. The role of leasing under adverse selection. J Polit Econ. 2002;110(1):113–43.

22. Johnson DS. Approximation algorithms for combinatorial problems. J Comput Syst Sci. 1974;9(3):256–78.

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

24. Lasfer MA, Levis M. The determinants of the leasing decision of small and large companies. Eur Financ Manag. 1998;4(2):159–84.

25. Li S, Markarian C, Meyer auf der Heide F. Towards flexible demands in online leasing problems. Algorithmica. 2018;80(5):1556–74.

26. Manne AS. Plant location under economies-of-scale-decentralization and competition. Manag Sci. 1964;11(2):213–35.

27. Markarian C. Leasing with uncertainty. In Operations Research Proceedings 2017, pp 429–434. Springer, 2018.

28. Markarian C. An optimal algorithm for online prize-collecting node-weighted steiner forest. In Combinatorial Algorithms - 29th
International Workshop, IWOCA 2018, Singapore, 16-19, 2018, Proceedings, pp 214 – 223, 2018.

29. Markarian C, Kassar A, Yunis MM. An algorithmic approach to online multi-facility location problems. In Parlier GH, Liberatore F, Demange M, editors, Proceedings of the 10th International Conference on Operations Research and Enterprise Systems, ICORES 2021, Online Streaming, 4-6, 2021, pp 29–35. SCITEPRESS, 2021.

30. Markarian C, Kassar A-N. Online deterministic algorithms for connected dominating set & set cover leasing problems. In International Conference on Operations Research and Enterprise Systems (ICORES), pp 121–128, 2020.

31. Markarian C, Khallouf P. Online facility service leasing inspired by the COVID-19 pandemic. In Gusikhin O, Nijmeijer H, Madani K, editors, Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2021, Online Streaming, 6-8, 2021, pages 195–202. SCITEPRESS, 2021.

32. Markarian C, Meyer auf der Heide F. Online algorithms for leasing vertex cover and leasing non-metric facility location. In Parlier GH, Liberatore F, Demange M, editors, Proceedings of the 8th International Conference on Operations Research and Enterprise Systems, ICORES 2019, Prague, Czech Republic, 19-21, 2019, pp 315–321. SciTePress, 2019.

33. Meyerson A. Online facility location. In Proceedings 42nd IEEE Symposium on Foundations of Computer Science, pp 426 – 431. IEEE, 2001.

34. Meyerson A. The parking permit problem. In 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS’05), pp 274–282. IEEE, 2005.

35. San Felice MC, Williamson DP, Lee O. The online connected facility location problem. In Pardo A, Viola A, editors, LATIN 2014: Theoretical Informatics, pp 574–585, 2014. Springer, Berlin.

36. Snyder LV. Facility location under uncertainty: a review. IIE Trans. 2006;38(7):547–64.

37. Snyder LV, Daskin MS. Stochastic p-robust location problems. IIE Trans. 2006;38(11):971–85.

38. Virkumar VV. Approximation Algorithms. Springer, New York, 2001.

39. Yan L, Chrobak M. Approximation algorithms for the fault-tolerant facility placement problem. Inf Process Lett. 2011;111(11):545–9.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.