Online Problems in Facility Location

by

Saeed Mehrabidavoodabadi

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
2012

Thesis advisor Author

Stephane Durocher **Saeed Mehrabidavoodabadi**

Online Problems in Facility Location

Abstract

We introduce two online models for the vertex k-center and the vertex k-median problems. Clients (i.e., graph vertices) and their corresponding links (i.e., graph edges) are revealed sequentially, determining the topology of a graph over time. Clients are revealed by an adversary to an online algorithm that selects existing graph vertices on which to open facilities; once open, a facility cannot be removed or relocated. We define two models: an online algorithm may be restricted to open a facility only at the location of the most recent client or at the location of any existing client. We examine these models on three classes of graphs under two types of adversaries. We establish lower bounds on the respective competitive ratios attainable by any online algorithm for each combination of model, adversary, and graph class. Finally, we describe algorithms whose competitive ratios provide corresponding upper bounds on the best competitive ratios achievable.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my supervisor, Dr. Stephane Durocher. Steph introduced the beautiful field of computational geometry to me. I learned from him how to explore potential research problems, how to apply different problem-solving techniques and how to present the results. Of course, I am still in the beginning. This work would have not been possible without his support, guidance and encouragement. Thanks Steph, I owe a huge debt to you!

This work was also supported by a Manitoba Graduate Scholarship and a computer science Guaranteed Funding Package. I would like to thank the members of the examining committee, Dr. Jason Morrison and Dr. Ben Pak Ching Li for their time and assistance. I would also like to thank my friends, Mehdi and his wife Jennifer, for their helps during my stay in Winnipeg.

Finally and most importantly, I would like to express my love and gratitude to my parents for their endless love and all encouragements throughout my life. My dear parents, this thesis is dedicated to you!

تقدیم به

پدر و مادر عزیزم!

Chapter 1

Introduction

1.1 Motivation

Suppose that the shopping company X opens two new branches in a city. The company locates the new branches such that its customers do not travel too long to access the nearest shopping center. Due to an increase in the number of customers, the company decides to open another third branch after the first year. Where is the best location to open the new branch? What strategy should company X adopt to select locations for its new branches in the future?

One approach might be to consider all customers in each year i as a static problem instance and find optimal locations for all shopping centers that the company X wants to have in the year i. However, in this case, we might be forced to relocate some shopping centers opened in previous years. A typical constraint is that once the company X opens a new branch, it cannot be moved in the future. We need a strategy to locate future branches without relocating existing branches such that the selected locations are close to the optimal strategy.

Motivated by such problems, we consider two new online models for the facility location problem. We first define the offline version of the problems.

In this thesis, we study facility location problems from the theoretical point of view and, therefore, we consider the input and output of a facility location problem as points in the plane. However, since the most common application of facility location problems is in communication networks, we sometimes refer to *clients* and *facilities* as the input and the output of a facility location problem, respectively.

Suppose that we are given a set P of n points (as clients) in a metric space M. In the k-center problem we wish to find a set F of k points in the plane (as facilities) so as to minimize

$$\max_{p \in P} \min_{x \in F} \operatorname{dist}_M(p, x), \tag{1.1}$$

where $\operatorname{dist}_M(p, x)$ denotes the distance between points p and x under the distance metric M. If M is Euclidean space under the Euclidean distance metric, then the problem is called the Euclidean k-center problem. If points are restricted to be selected only from the clients in P (i.e., $F \subseteq P$), then we call the problem the discrete Euclidean k-center problem. Otherwise, if points can be selected from anywhere in Euclidean space, then we call the problem the continuous Euclidean k-center problem.

Now, assume that the clients model a communication network whose configuration is determined by some (unweighted) graph $G_P = (V, E)$, where V is the set of clients and E is a set of predefined connections between the clients. Then, if M is the graph distance on G_P and $F \subseteq V$, then the new problem is called the vertex k-center problem. Recently, Durocher et al. [11] introduced the geometric k-center problem:

suppose both the clients set P and facilities set F are points in the plane and M is graph distance on some geometric graph induced by P and F (e.g., a unit disk graph). The objective is the same as the two previous problems, that is, to minimize (1.1).

In the k-median problem, on the other hand, we wish to find such a set F to minimize

$$\sum_{i=1}^{n} \min_{f \in F} \mathrm{dist}_M(c_i, f), \tag{1.2}$$

where $\mathrm{dist}_M(c_i, f)$ is the distance between the client c_i and the facility f based on the distance metric M. Observe that (1.1) minimizes the maximum client-to-facility distance whereas (1.2) minimizes the sum of distances between clients and their nearest facility, which equivalently minimizes the average client-to-facility distance.

In this thesis, we study new online models for two facility location problems, namely *the vertex k-center* and *the vertex k-median* problems. We describe the models and the settings under which we study them in Chapter 2.

1.2 Online Algorithms

An online algorithm is one that receives the input step by step in a number of increments. An online algorithm does not know the entire input sequence in advance and is required to perform its actions based on only the current partial information. In contrast, an offline algorithm receives the entire input at first and has the complete knowledge of the input in advance.

Since an online algorithm computes a solution using partial information about the input, the performance of the algorithm is reduced in comparison with that of an offline algorithm. Competitive analysis is the standard measure to analyze the performance of an online algorithm [4]. Let $OPT(\sigma)$ denote the cost incurred by an optimal offline algorithm on input σ (i.e., one that knows the entire input σ in advance and performs its actions optimally). An online algorithm A is said to be c-competitive if

$$A(\sigma) \leq c \cdot OPT(\sigma) \tag{1.3}$$

for all input sequences σ [10]. The value c is called the competitive ratio of A. For further discussion of online algorithms see [4]. Dorrigiv [10] is a good reference for the different measures of online algorithms and their comparisons.

1.3 Related Work

We first present a brief overview of work related to the offline k-center problem and then will describe previous research related to the online k-center problem. Finally, results on the offline Euclidean k-median problem are presented.

1.3.1 Vertex k-Center Problem

In the vertex k-center problem, clients are represented as a set of vertices V, the distance metric is defined by graph distance in a graph $G_P = (V, E)$, where $n = |V|$ and $m = |E|$, and the facilities are restricted to be opened only on clients. Let n be the number of clients (or equivalently, the number of vertices of the graph). Frederickson [13] solves this problem for trees in $O(n)$ using parametric search. Kariv and Hakimi [19] define algorithms for the vertex k-center problem on general graphs. Since the problem is NP-complete [19], they give an algorithm with running time

$O(m^k n^{2k-1}/(k-1)!)$ when k is fixed.

Bespamyatnikh et al. [3] give an $O(kn)$ time algorithm to solve the problem on circular-arc graphs. Since the vertex k-center problem on unit interval graphs is a special case of circular-arc graphs, therefore, we can solve the vertex k-center problem on unit interval graph in $O(kn)$ time. Cheng et al. [9] improve this time and design an $O(n)$ time algorithm to solve the vertex k-center problem on unit interval graphs.

1.3.2 Continuous Euclidean k-Center Problem

When k=1, the continuous Euclidean k-center problem is equivalent to finding the minimum enclosing disk for n points in the plane. Megiddo [22] presents a deterministic $\Theta(n)$ time algorithm for the minimum enclosing disk problem. Therefore, the Euclidean 1-center problem in the plane can be solved in linear time.

Chan [6] gives a deterministic algorithm with running time $O(n \log^2 n \log^2 \log n)$ for the continuous Euclidean 2-center problem in the plane. Agarwal and Sharir [1] extend the continuous Euclidean 2-center problem to higher dimensions. When k is an input parameter, Megiddo and Supowit [23] prove that the continuous Euclidean k-center problem is NP-hard in the plane. Feder and Greene [12] show that approximating the problem in polynomial time within a factor less than $(1+\sqrt{7})/2$ is still NP-hard. Gonzalez [14] provides a 2-approximation algorithm for the continuous Euclidean k-center problem in the plane. His algorithm works in k iterations. In the first iteration, an arbitrary client is chosen to own the first facility. In the subsequent iterations, a facility is opened on the location of that client that has the maximum distance to its nearest facility. He shows that the running time of the algorithm is bounded by $O(nk)$.

1.3.3 Discrete Euclidean k-Center Problem

In the discrete Euclidean k-center problem, facilities are restricted to be opened only on the locations of existing clients. As described in Section 1.3.2, Gonzalez' algorithm works even if the facilities are restricted to be opened only on the locations of clients. Therefore, we can apply his algorithm for the discrete Euclidean k-center problem. Hsu and Nemhauser [16] show that the discrete Euclidean k-center problem cannot be approximated within a factor of less than 2.

1.3.4 Geometric k-Center Problem

In the geometric k-center problem, the facilities can be opened anywhere in the plane, however, the distance metric is graph distance. Durocher et al. [11] design and analyze the first exact and approximate algorithms for the geometric k-center problem. We denote by m and n, the number of vertices and edges of the unit disk graph induced by P. Two vertices are adjacent in a unit disk graph if and only if whose distance is at most one. First, they use the vertex k-center problem to solve the proposed problem. They prove that an algorithm for the vertex 1-center problem provides a 5-approximation algorithm for the geometric 1-center problem. Second, they use breadth-first search and describe an $O(mn^2)$ time algorithm to find exact solutions for the geometric 1-center problem. Then, they describe an $O(n^3)$ time algorithm that finds approximate solutions for the geometric 1-center problem. To generalize, they use their previous algorithm for exact solutions to design an $O(mn^{2k})$

algorithm for the geometric k-center problem, for any fixed value k. Finally, they prove that the geometric k-center problem is NP-hard in the plane, when k is an input parameter.

1.3.5 Online Center Problem

There are few variations related to the online k-center problem. Sharp [24] describes the online center problem. The online center problem has the same input as the k-center problem, however, the value of k is not known in advance. Instead, k is a parameter that increases as time passes. Thus, the algorithm has advance knowledge of all client positions. In fact, the online center problem is online with respect to facilities but not clients. Therefore, the 2-approximation algorithm of Gonzalez [14] is a 2-competitive algorithm for the online center problem.

Lin et al. [21] consider the incremental center problem in which the parameter k is known in advance but the problem is not the same as the offline k-center problem. The objective in the incremental center problem is to locate a sequence of k facilities such that the radius of the underlying graph after opening the first i facilities is close to that of the optimal i facilities, for each $i \leq k$. Again, Gonzalez' 2-approximation algorithm is a 2-competitive incremental algorithm, which is optimal. This means that, in this case, knowing the input sequence in advance will not improve the solution.

1.3.6 Continuous Euclidean k-Median Problem

Despite its simple description, the k-median problem is challenging. Even when $k = 1$, there exists no polynomial-time exact algorithm for the Euclidean k-median problem in $\mathbb{R}^2$ and it is unknown if the problem is NP-hard [15]. Chandrasekaran and Tamir [7] give a polynomial-time ϵ-approximation algorithm for the Euclidean 1-median problem. Indyk [17] presents a randomized ϵ-approximation algorithm with expected linear time. Finally, Bose et al. [5] give an deterministic $O(n \log n)$-time ϵ-approximation algorithm for this problem.

For $k > 1$, it is natural that the k-median problem is harder than the 1-median problem. When k is an input parameter, Megiddo and Supowit [23] show that the Euclidean k-median problem is NP-hard in $\mathbb{R}^2$. Jain and Vazirani [18] give an $O(n^2)$-time 6-approximation algorithm and Charikar and Guha [8] give an $O(n^3)$-time 4-approximation algorithm for the k-median problem. Arora et al. [2] give an $O(n^{O(1+1/\epsilon)})$-time ϵ-approximation algorithm, for any fixed $\epsilon > 0$. Finally, Killiopoulos and Rao [20] give a randomized approximation scheme for points in d-dimensional space with running time $O(2^{1/\epsilon d} n \log n \log k)$ for any fixed ϵ and d.

Chapter 2

Problem Statement

In this chapter, we first state the problem and describe our two online models. Section 2.2 lists the settings under which we study the models of the problem. We present an overview of the results in Section 2.3.

We remind the reader that we study two online models of the vertex k-center and the vertex k-median problems. Throughout the thesis, we use *the online k-center* and *the online k-median* to refer to the online version of the vertex k-center and the vertex k-median problems, respectively.

2.1 Description of Models

Suppose a set P of n clients appears in an online fashion and let F denote a set of k facilities to serve clients. Recall that in the vertex k-center (and similarly the vertex k-median) problem the facilities can only be opened on the location of clients. We introduce a discrete temporal dimension $T = \{1, 2, \ldots, n\}$ to the sets of clients and facilities. Let P_i, $1 \leq i \leq n$, be the set of clients that exist at time i. That is

$$P_i = \begin{cases} \emptyset, & \text{if } i = 0, \\ P_{i-1} \cup \{c_i\}, & \text{if } i > 0, \end{cases} \tag{2.1}$$

where c_i, $1 \leq i \leq n$, is the client that arrives at time i. Let F_i, $1 \leq i \leq n$, denote the set of facilities opened for P_i, where $1 \leq |F_i| \leq k$. Thus, for all i, $P_i \subset P_{i+1}$ and $F_i \subseteq F_{i+1}$. We simply write P and F instead of P_n and F_n, respectively. Recall the definition of G_P from Section 1.1; the graph determined by the set of clients in the network. In the online settings we generalize this notion by letting G_{P_i} denote the graph determined by clients in P_i. Depending on the model (see Section 2.2), the competitive ratio is measured either every time a new facility is opened or only when the last client arrives; let $T' \subseteq T$ denote the corresponding times and let M be the distance metric.

The objective in the online k-center problem is to minimize

$$\max_{i \in T'} \max_{p \in P_i} \min_{f \in F_i} \mathrm{dist}_M(p, f), \tag{2.2}$$

where $\mathrm{dist}_M(p, f)$ is the distance between the client p and the facility f in G_{P_i}.

For the online k-median problem, we observe that the sum of distances between existing clients and their nearest facilities never decreases after the arrival of a new client. Thus, the objective of the online k-median problem is analogous to that of the offline k-median problem; that is, to minimize (1.2). Therefore, in this problem we measure the objective after the algorithm's execution (i.e., after the arrival of c_n).

At first, there is no client in the network. As time passes, the clients appear one at a time in some (possibly adversarial or random) order. On the arrival of each client, consider G_{P_i}. While we have no knowledge of the exact location of future clients, we assume that their appearance never makes G_{P_i} disconnected. Moreover, on the arrival of a client c, every edge between c and its existing neighbors are revealed to the algorithm. Note that, there is at least one such edge because the graph G_{P_i} must be connected. The two online models are as follows:

Most Recent Client (MRC) : On the arrival of c, the algorithm decides whether to open a new facility on c or c will be served by its nearest existing facility.

Any Existing Client (AEC) : On the arrival of c, the algorithm decides whether to open a new facility on the location of an existing client. If the algorithm opens a facility on the location of some client other than c, then c will be served by its nearest existing facility.

Each client must be served by some facility at all times. Consequently, in both models above, any algorithm must open the first facility on the location of the first client because no facility is open yet. Once an algorithm opens a facility, the facility's location remains fixed and cannot be moved in future.

We use graph distance as the distance metric in all of our results. Consider the graph G_{P_i}. The *radius* of G_{P_i} is defined as the maximum distance between any client and its nearest facility. We observe that the objective in the k-center problem (see (2.2)) is to minimize the radius of G_{P_n}. Moreover, the *diameter* of the graph is the maximum distance between any two clients in the graph.

2.2 Settings

In this section, we first describe how we measure the competitive ratio in each model. Then, we describe the relationship between the number of clients and the number of facilities. Finally, we describe the classes of graphs under which we study the models.

2.2.1 Measuring the Competitive Ratio

We measure the competitive ratio for each k-center and k-median problem in a different way. We first introduce a set $P' \subseteq \{P_1, P_2, \ldots, P_n\}$ of checkpoints such that we measure the value of the objective function achieved by an algorithm on G_{P_i} for all $P_i \in P'$. Let $P_j \in P'$ be the client set for which the objective function is maximum over all client sets in P'. Then, we compare the value of the objective function achieved by the algorithm on G_{P_j} with that of an optimal offline algorithm on G_{P_n}.

In the online k-center problem, we measure the competitive ratio of an online algorithm differently depending on the model. In the MRC model, we observe that a good online algorithm does not postpone opening the facilities until the arrival of the last clients because then it cannot open facilities on the location of previous clients, resulting in a bad performance. Naturally, a good online algorithm for the MRC model is one that opens facilities uniformly distributed among clients over time. Thus, for the MRC model, $P' = \{P_n\}$; that is, it is sufficient to measure the objective function achieved by the algorithm only at the end of the execution (i.e., after the arrival of c_n, the last client). Therefore, we do not measure the value of the objective function at intermediate times.

For the AEC model, $P' = \{P_{i_1}, P_{i_2}, \ldots, P_{i_k}\}$ such that $\{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$ is the set of clients for which the algorithm opens a facility upon their arrival.

In the online k-median problem, the objective is to minimize (1.2); that is, the sum of the client-to-facility distances. We can use the same argument as for the online k-center problem under the MRC model to show that it is sufficient to have $P' = \{P_n\}$ for measuring the competitive ratio of an algorithm for the online k-median problem under the MRC model.

2.2.2 The Number of Clients

We study the models under two settings with respect to the number of clients. In the first setting, the number of clients is independent of the number of facilities. This means that the adversary can provide an algorithm with an arbitrary number of clients. We call this variant the *strong adversary*. In the second setting, we assume that the number of clients is linear in the number of facilities, i.e., $n \in O(k)$. We call this variant the *linear adversary*. Under all models we assume that n and k are known to the algorithm.

2.2.3 Graph Representation

As we mentioned earlier, we always assume that for all $i \in \{1, 2, \ldots, n\}$, (i) the graph G_{P_i} is connected, and, (ii) G_{P_i} is a supergraph of $G_{P_{i-1}}$. We study the two models under different graph representations of clients, namely:

Paths. We assume that clients arrive all on a line such that each client is located only to the left or to the right of the all existing clients.

Trees. In this setting, we assume that the underlying graph obtained by clients is a tree. Each new client is added as a new leaf on the existing tree.

General Graphs. Clients arrive anywhere and there is no restriction on the configuration of the underlying graph. However, the graph must remain connected at all times.

Geometric Graphs. In the following, we briefly describe another application that is related to our models, however, we require a modification of a constraint in our models to capture the situation. Suppose that G_{P_i} represents a model for a wireless network where clients join one at a time. Since there is no wired communication, assume that each client is equipped with a transceiver, which can send or receive data. The message sent by some client c can only be received by those clients that are within a given distance of c, where the distance is proportional to the transmission power of c. Geometrically, the region of transmission corresponds to a disk. It is common to assume the all nodes have equal transmission radius (e.g., one unit). In most applications the underlying *unit disk graph* must be connected. In other words, once a new client arrives in the network, it must be able to communicate with at least one existing client.

The above scenario motivates the study of our online models under the geometric graphs. However, the connectivity between the vertices of a geometric graph is determined by position of the vertices of the graph while the connectivity in our models is defined by an adversary. Thus, there might be two non-adjacent clients in the network with distance less than a unit length. Moreover, recall that in our models we

Table 2.1: The results of the online k-center problem under the MRC model.

Adversary:	Linear		Strong	
Graph	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Paths	2	2	2	2
Trees	2	$\frac{2n}{k}$	$2k-1$	$3k$
General Graphs	2	$\frac{2n}{k}$	$2k-1$	$3k$

require that $P_i \subset P_{i+1}$, $0 \leq i < n$. Therefore, we cannot study the online models for all proximity graphs (e.g., *Gabriel graphs*, *relative neighborhood graphs*, *nearest neighbor graphs*), because in many proximity graphs the existing edges may be replaced with new edges once a new vertex is added. One possible way to enable the models to capture this situation is to restrict the adversary to only choose the location of clients (and not their connectivity), and then the connectivity is determined by the definition of the geometric graph.

2.3 Organization and Overview of Results

The remainder of this thesis is organized as follows.

Chapter 3. In this Chapter we present our results for the online k-center problem under the MRC model. We first observe the problem for paths and provide tight bounds on the competitive ratio of any online algorithm. Then, we give the results for trees and general graphs. Table 2.1 summarizes the results for the online k-center problem under the MRC model.

Chapter 4. This Chapter examines the online k-center problem under the AEC model. Similar to the MRC model, we start studying the problem for paths. Based

Table 2.2: The results of the online k-center problem for the AEC model.

Adversary:	Linear		Strong	
Graph	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Paths	2	2	2	2
Trees	2	$\frac{2n}{k}$	$\frac{2(k-1)}{3}$	$3k$
General Graphs	2	$\frac{2n}{k}$	$\frac{2(k-1)}{3}$	$3k$

Table 2.3: The results of the online k-median problem for the MRC model.

Adversary:	Linear		Strong	
Graph	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Paths	$\frac{2k}{2k-1}$	$1+\frac{1}{k}$	$\frac{2k}{2k-1}$	$1+\frac{1}{k}$
Trees	$\Omega(\sqrt{m})$	$O(\frac{n^2}{k})$	$\Omega(k)$	?
General Graphs	$\Omega(\sqrt{m})$	$O(\frac{n^2}{k})$	$\Omega(k)$	?

on the simple structure, there are similarities between the MRC and AEC models for paths. Using these similarities, we also give tight bounds under this model. Next, we study the model using trees and general graphs. Table 2.2 describes our results for this model.

Chapter 5. In this Chapter, we explore the online k-median problem under the MRC model. As in the offline versions, the online k-median problem is more difficult than the online k-center problem. Table 2.3 summarizes our results for the online k-median problem under the MRC model.

Chapter 6. We conclude the thesis in this Chapter and discuss possible directions for future work.

Chapter 3

The Online k-Center Problem: The MRC Model

In this chapter, we investigate the MRC model of the online k-center problem under both strong and linear adversaries. We begin by defining some notation. Then, in each of the subsequent sections, we consider one of the graph representations for both strong and linear adversaries.

3.1 Preliminaries

Recall G_{P_i}, the graph induced by clients in P_i. Let A be an online algorithm that has opened some facilities after the arrival of the ith client, c_i. We denote the radius achieved by A on G_{P_i} by $\text{rad}(A, G_{P_i})$. Moreover, throughout this chapter, we denote an optimal offline algorithm by OPT. We use CR to denote the competitive ratio achieved by a given online algorithm on any input sequence.

3.2 Paths

We first consider the MRC model for paths. Note that, in this case, G_{P_i} is a path with i vertices. The strong and linear adversaries are similar for paths in this model because the difference between the diameters of G_{P_i} and $G_{P_{i+1}}$ is exactly one, for all $0 \leq i < n$. We use a linear adversary in the following result, which implies a corresponding lower bound for a strong adversary.

Theorem 3.2.1. *There is a lower bound of 2 on the competitive ratio of any online algorithm for the MRC model against a linear adversary.*

Proof. Consider an adversarial strategy for positioning clients online. In other words, given a value of k, we construct a path with $2k+1$ vertices such that any online algorithm has competitive ratio of at least 2 on the path. We start by positioning clients from left to right in an online fashion such that each two consecutive clients are adjacent. We denote by c_i the client at position i and arrival time i. Let A be an online algorithm.

Suppose that there exist two consecutive clients c_i and c_{i+1}, for some $i \in \{2, 4, \ldots, 2(k-1)\}$, neither of which has a facility.[1] Now, we will position c_{i+2} to the left of the first client and then will position the all remaining clients to the left of c_{i+2} from right to left. Note that in this case, $\text{rad}(A, G_{P_n}) \geq 2$ while $\text{rad}(OPT, G_{P_n}) = 1$. Thus, CR $\geq 2/1 = 2$.

Now, suppose that the first case does not happen; that is, at least one of c_i and c_{i+1} has a facility, for all $i = 2, 4, \ldots, 2(k-1)$. Since A has opened the first facility at c_1, we conclude that A has no more facility to open after the arrival of c_{2k}. Now,

[1] Note that A must open the first facility at c_0.

we position the last client neighbor to c_{2k}. Clearly, A has not opened any facility at clients c_{2k} and c_{2k+1}. In this case, OPT opens the facilities at clients c_1, c_3, $\ldots$, c_{2k-1} and, therefore, we have $\textsc{cr} \geq 2/1 = 2$. $\square$

In the following, we present a 2-competitive algorithm for this problem.

Algorithm. The first facility is opened on the location of c_1. On the arrival of the client c_i, $i \geq 2$, if $\mathrm{rad}(A, G_{P_i}) \geq \lfloor n/k \rfloor$, then we open a facility at c_i. We call this algorithm MRCPathAlgorithm, denoted by A_1.

Analysis. First, note that the algorithm never runs out of facilities because the number of times that the radius can exceed the threshold $\lfloor n/k \rfloor$ is at most $k-1$ in the worst case.[2] Second, let c_i, for some $2 \leq i \leq n$, be a client for which algorithm A_1 opens a facility. Therefore, c_i is the client that makes the radius of G_{P_i} greater than the threshold $\lfloor n/k \rfloor$. Therefore, the radius of P_i reduces below the threshold $\lfloor n/k \rfloor$ by opening a facility at c_i. By these two observations, we conclude that $\mathrm{rad}(A_1, G_{P_n}) \leq n/k - 1$. Since $\mathrm{rad}(OPT, G_{P_n}) \geq \lfloor (n-k)/2k \rfloor$ the competitive ratio of A_1 follows.

We observe that Algorithm A_1 works even against a strong adversary. Therefore, we have the following theorem.

Theorem 3.2.2. MRCPathAlgorithm *is a 2-competitive algorithm for the MRC model of the online k-center problem for paths against both linear and strong adversaries.*

[2] Otherwise, the diameter of P_n will be greater than $n-1$, which is impossible for a path with n vertices.

3.3 Trees and General Graphs

In this section, we present our results for both trees and general graphs. We explore both classes of graphs in the same section because of the following observation.

Observation 3.3.1. *Given an online algorithm A, a lower bound on the competitive ratio of A for trees is also a lower bound on the competitive ratio of A for general graphs. Similarly, an upper bound on the competitive ratio of A for general graphs is also an upper bound on the competitive ratio of A for trees.*

3.3.1 Strong Adversary

We first consider the MRC model against a strong adversary and provide a lower bound of $2k-1$ on the competitive ratio of any online algorithm for trees.

Theorem 3.3.2. *For any online algorithm A, there exists some sequence of clients revealed by a strong adversary whose graph G_{P_n} is a tree, and for which the competitive ratio of A is at least $2k-1$.*

Proof. Given a fixed value of k, we construct a tree G incrementally. We divide the arrival of clients into at most k steps. Let $\mathcal{I}_i$, $1 \leq i \leq k$, denote the sequence of clients arrived in step i. We keep providing $\mathcal{I}_i$ until, depending on the decisions made by A, the radius achieved by A on the underlying graph is at least $2k-1$. Suppose that clients are revealed to A in an online fashion.

Step 1. First, we reveal $\mathcal{I}_1$, which contains $3k$ clients all on a line and any two clients c_i and c_{i+1}, $1 \leq i \leq 3k-1$, are adjacent to each other, see Figure 3.1. We call

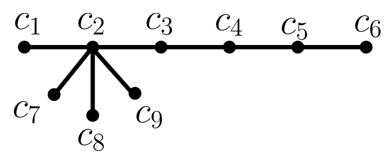

Figure 3.1: An example of graph G with $k = 2$ and $n = 9$.

this partial graph G.[3] Note that, while the all graphs constructed in this proof are trees, we refer to them all as graphs. There are two possibilities:

Case 1: Algorithm A does not open any other facility up to c_{3k}. Now, we position all remaining clients adjacent to c_2 (see Figure 3.1) and we have

$$\text{CR} = \frac{\text{rad}(A, G_{P_n})}{\text{rad}(OPT, G_{P_n})} = (3k - 1)/1 = 3k - 1 > 2k - 1.$$

The construction stops here and does not proceed to Step 2.

Case 2: Algorithm A opens at least one more facility, in addition to the first one, at c_{t_1} where $2 \leq t_1 \leq 3k$. Then, we move to Step 2.

Step 2. In this step, we reveal the following sequence of clients in addition to clients in the first step:

$$\mathcal{I}_2 = \langle \mathcal{I}_1, \mathcal{I}_1, \ldots, \mathcal{I}_1 \rangle,$$

where $\mathcal{I}_2$ contains $2k - 1$ copies of input sequence $\mathcal{I}_1$. Consider the graph H in Figure 3.2. We show each instance of the graph G by a square, called *block*. Each block contains exactly $3k$ clients. Specifically, the first block is actually the graph G itself.

We observe that there exists one more client after the arrival of the last client in every odd block (i.e., first block, third block and so on) but there is no such

[3] Note that the graph G is different from graph G_{P_i}, for all $0 \leq i \leq n$.

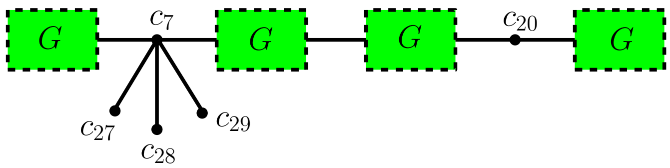

Figure 3.2: An example of graph H with $k = 2$ and $n = 29$.

client appeared after the arrival of the clients in an even block (second block, forth block and so on). There are two cases:

Case 1: Algorithm A does not open a facility on some client that arrived in this step. Then, we position the all remaining clients adjacent to c_{3k+1}; see Figure 3.2. Thus, we have

$$\text{rad}(A, G_{P_n}) \geq 2(k-1)3k + 3k + k - 1 = (2k-1)3k + k - 1.$$

Thus,

$$\begin{aligned} \text{CR} &\geq \frac{\text{rad}(A, G_{P_n})}{\text{rad}(OPT, G_{P_n})} = (3k(2k-1) + k - 1)/3k \\ &= 2k - 1 + \frac{k-1}{3k} \geq 2k - 1. \end{aligned}$$

The construction stops here and does not proceed to Step 3.

Case 2: Algorithm A opens at least one facility at some client that arrived in this step; more precisely at c_{t_2}, where $3k + 1 \leq t_2 \leq (2k)3k + k$. Then, we proceed to the next step.

Step 3. In this step, we reveal the following sequence of clients to Algorithm A in addition to clients in the previous steps:

$$\mathcal{I}_3 = \langle \mathcal{I}_2, \mathcal{I}_2, \ldots, \mathcal{I}_2 \rangle,$$

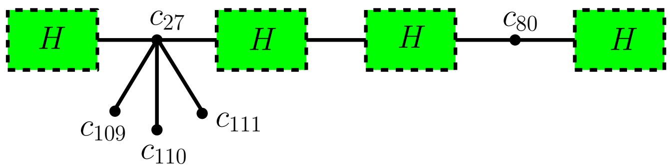

Figure 3.3: An example of graph I with $k = 2$ and $n = 111$.

where $\mathcal{I}_3$ contains $2k-1$ copies of input sequence $\mathcal{I}_2$. Note that, in this step, we have $2k-1$ instances of graph H instead of graph G. See Figure 3.3. We show each instance of the graph H by a square, namely a *block*. Again, there is a client between every odd block and its next block but there are no more clients between any even block and its next block. There are two choices:

Case 1: Algorithm A does not open a facility at some client arrived in this step. More precisely, Algorithm A does not open any facility at c_{t_3}, where

$$6k^2 + k + 1 \leq t_3 \leq (2k)[(2k)3k + k] + k.$$

Then, as Figure 3.3 shows, we position all remaining clients adjacent to $c_{(2k)3k+k+1}$ and we have

$$\begin{aligned} \text{rad}(A, G_{P_n}) &\geq 2(k-1)[(2k)3k + k] + (2k)3k + k - 1 \\ &= (2k-1)[(2k)3k + k] + k - 1. \end{aligned}$$

Furthermore, $\text{rad}(OPT, G_{P_n}) = (2k)3k + k$. Therefore,

$$\begin{aligned} \text{CR} &\geq \frac{\text{rad}(A, G_{P_n})}{\text{rad}(OPT, G_{P_n})} \\ &= = \frac{(2k-1)[(2k)3k + k] + k - 1}{(2k)3k + k} \\ &= (2k-1) + \frac{k-1}{(2k)3k + k} \geq 2k - 1. \end{aligned}$$

Again, the construction stops here and does not proceed to the next step.

Case 2: Algorithm A opens at leat one facility at some client arrived in this step. Then, we move to the next step.

We continue the steps until one of the followings happen:

1. During some step i, $3 < i < k$, Algorithm A does not open any new facility. Now, to compute CR in Step i, we introduce the following recurrence relation:

$$T(i) = \begin{cases} 1, & \text{if } i = 0 \\ 2kT(i-1) + k, & \text{if } i > 0 \end{cases} \tag{3.1}$$

   Note, $T(i) > 0$ for all $i \geq 0$. If no facility was opened during step $i \geq 1$, then $\mathrm{rad}(A, G_{P_n}) \geq (2k-1)T(i-1) + k - 1$ while $\mathrm{rad}(OPT, G_{P_n}) = T(i-1)$. Therefore,

$$\begin{aligned} \text{CR} &= \frac{\mathrm{rad}(A, G_{P_n})}{\mathrm{rad}(OPT, G_{P_n})} \\ &\geq \frac{(2k-1)T(i-1) + k - 1}{T(i-1)} \\ &= 2k - 1 + \frac{k-1}{T(i-1)} \\ &\geq 2k - 1. \end{aligned}$$

   The last inequality holds because $T(i) > 0$ for all $i \geq 0$ and, therefore, $\frac{k-1}{T(i-1)} \geq 0$.

2. We found no such i in the previous case. Then, we continue to Step $i = k$, where Algorithm A has no facility to open. We position the sequence $\mathcal{I}_k = \langle \mathcal{I}_{k-1}, \mathcal{I}_{k-1}, \ldots, \mathcal{I}_{k-1} \rangle$ of clients, where $|\mathcal{I}_k| = 2k - 1$. Since Algorithm A has

no facility to open for the clients within Step k and by the discussion in the previous case, we conclude that CR $\geq 2k-1$. This completes the proof. □

The lower bound $2k-1$ given in Theorem 3.3.2 also applies to general graphs. Next, we give a simple online algorithm with competitive ratio of $3k$ for general graphs.

Algorithm. The algorithm opens the first facility on the location of c_1. Then, on the arrival of client c_i, $i \geq 2$, if $\text{dist}(c_i, f) \geq 2n/k$, where f is the nearest facility to client c_i, then we open a new facility at c_i. We call this algorithm StrongMRC, denoted by A_2.

Analysis. It is straightforward that Algorithm A_2 does not run out of facilities. Therefore, $\text{rad}(A_2, G_{P_n}) \leq 2n/k$. If $\text{diam}(G_{P_n}) \geq 4n/3k + k$, then $\text{diam}(G_{P_n}) - k \geq 4n/3k$. Thus,

$$\text{CR} = \frac{\text{rad}(A_2, G_{P_n})}{\text{rad}(OPT, G_{P_n})} \leq \frac{2n/k}{(\text{diam}(G_{P_n}) - k)/2k} \leq \frac{4n}{4n/3k} = 3k.$$

Now, suppose that $\text{diam}(G_{P_n}) < 4n/3k + k$. Assume that CR $> 3k$. Thus, there exists an instance of the problem for which $\text{rad}(A_2, G_{P_n}) / \text{rad}(OPT, G_{P_n}) > 3k$. Let $\text{rad}(OPT, G_{P_n}) = t$. Therefore, there is a client c whose distance to its nearest facility opened by OPT in graph G_{P_n} is t. Moreover, we conclude that there exists a client c' whose distance to its nearest facility opened by Algorithm A_2 in graph G_{P_n} is greater than $3kt$. Thus, there is a path of length greater than $3kt$ in graph G_{P_n}. This contradicts the fact that $\text{rad}(OPT, G_{P_n}) = t$ because, since $\text{rad}(OPT, G_{P_n}) = t$, the maximum length for a path in G_{P_n} is $(2t+1)k \leq 3kt$. Therefore, CR $\leq 3k$.

By Observation 3.3.1, this algorithm is also a $3k$-competitive algorithm for trees. Therefore, we obtain the following theorem:

Theorem 3.3.3. *There exists an online algorithm with competitive ratio of at most $3k$ against a strong adversary under the MRC model of the online k-center problem for both trees and general graphs.*

3.3.2 Linear Adversary

In this section, we consider the MRC model for trees and general graphs against a linear adversary. Theorem 3.2.1 provides a lower bound of 2 on the competitive ratio of any online algorithm for trees and general graphs against a linear adversary. In the following, we present a $(2M)$-competitive algorithm, where $M = n/k$, for general graphs against a linear adversary.

Algorithm. The algorithm opens the first facility on the location of c_1. Then, on the arrival of client c_i, $i \geq 2$, if $\text{dist}(c_i, f) \geq n/(k-1)$, where f is the nearest facility to client c_i, then we open a new facility at c_i. We call this algorithm LinearMRC, denoted by A_3.

Analysis. Since the algorithm always has facilities to open, $\text{rad}(A_3, G_{P_n}) \leq n/(k-1) \leq 2n/k = 2M$. Therefore,

$$\text{CR} = \frac{\text{rad}(A_3, G_{P_n})}{\text{rad}(OPT, G_{P_n})} \leq \frac{2M}{(\text{diam}(G_{P_n}) - k)/2k} = \frac{4n}{\text{diam}(G_{P_n}) - k}$$

Case 1. If $\text{diam}(G_{P_n}) \geq 3k$, then $\text{CR} \leq 4n/(\text{diam}(G_{P_n}) - k) \leq 4n/(3k - k) = 2M$.

Case 2. If $\text{diam}(G_{P_n}) < 3k$. Suppose, by a contradiction, that $\text{CR} > 2M$. Moreover, let $\text{rad}(OPT, G_{P_n}) = t$; thus, there is a client c whose distance to its nearest facility opened by OPT in graph G_n is t. Therefore, there exists a client c' whose distance to its nearest facility opened by A_3 in graph G_{P_n} is greater than $2Mt$. In the following, we prove that $2Mt \geq n/(k-1)$ assuming that $k \geq 2$. This is a contradiction because the algorithm had to open a facility on client c' at the time of its arrival.

To prove the inequality, we observe that if $k = 1$, then the problem becomes trivial as any algorithm has to open the facility on the location of the first client resulting in $\textsc{cr} \leq 2$. If $k \geq 2$, then $2nk - 2n \geq 2nk - nk$. Thus, $(2nk - 2n)t \geq (2nk - nk)t$ as $t \geq 1$. But, $(2nk - nk)t = nkt \geq nk$. Therefore, $2nt(k-1) \geq nk$, which implies that $n/(k-1) \leq 2nt/k = 2Mt$.

Therefore, we have the following theorem:

Theorem 3.3.4. *Algorithm* LinearMRC *is a* $(2M)$*-competitive algorithm, where* $M = n/k$*, for the MRC model of the online* k*-center problem for trees and general graphs against a linear adversary.*

Chapter 4

The Online k-Center Problem: The AEC Model

In this chapter, we investigate the AEC model of the online k-center problem. In each of the subsequent sections, we consider one of the graph representations for both strong and linear adversaries. Throughout this chapter, we use the same notation as defined in Section 3.1.

An algorithm for the AEC model can potentially achieve a better competitive ratio than that of an algorithm for the MRC model. To see this, let $k = 2$ and consider the sequence of clients that form the graph shown in Figure 4.1(a). Moreover, let A (resp., B) be an online algorithm for the MRC (resp., the AEC) model. Both A and B must open the first facility on the location of the first client. If Algorithm A does not open a facility on the location of any of the clients c_2 through c_i, then an adversary may open all remaining clients adjacent to client c_i (as shown in Figure 4.1(a)). Algorithm A can never open a facility on clients c_2 through c_i. Algorithm B, however, can open

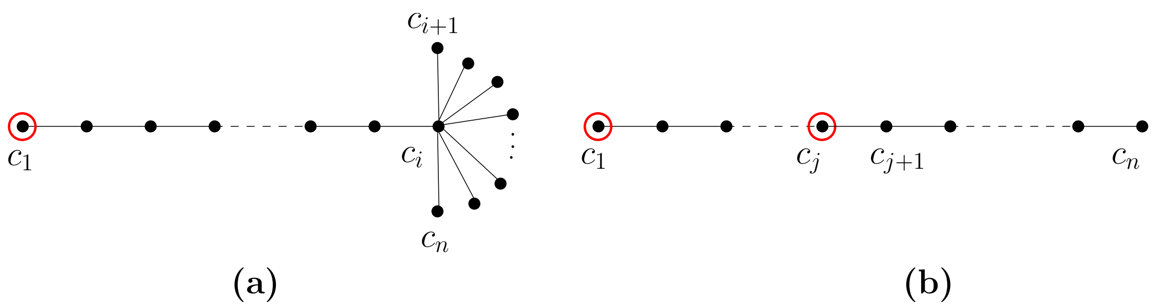

Figure 4.1: An algorithm for the AEC model is more powerful than an algorithm for the MRC model.

a facility on the location of any existing client at any time. Therefore, Algorithm B can achieve a smaller radius relative to Algorithm A, resulting in a better competitive ratio for Algorithm B. If Algorithm A opens a facility on the location of client c_j, for some $2 \leq j \leq i$, then an adversary may open all remaining clients on a path (as shown in Figure 4.1(b)). We also observe that in this case, Algorithm B can obtain a better competitive ratio by opening the second facility on client c_t, where $t = \lfloor 3n/4 \rfloor$.

Based on the scenario above, we conclude that an algorithm for the AEC model can achieve better competitive ratios relative to an algorithm for the MRC model. However, we were unable to find algorithms for the AEC model with better competitive ratios and, therefore, our algorithms are adopted from the MRC model.

Remark 4.0.5. *In this section, each of our algorithms decides whether to open a facility depending on whether the most recent client causes the radius to increase above some given threshold. Each new facility opened by the algorithm decreases the radius below the threshold, implying that our algorithms never require opening more than one facility at a time. The new facility can be located on any existing client, potentially allowing a greater reduction in the radius than would have been possible under the MRC model.*

4.1 Paths

In this section, we consider the AEC model for paths. As in Chapter 3, note that the strong and linear adversary settings are similar for paths even in this model. We give a lower bound of 2 for this model in which we use a linear adversary. Therefore, the same lower bound also holds for strong adversary.

Theorem 4.1.1. *There is a lower bound of 2 on the competitive ratio of any online algorithm for paths under the AEC model of the online k-center problem.*

Proof. We describe an adversarial strategy for defining client positions. Let $k > 0$, the number of facilities, be an even integer and consider any online algorithm A. We denote the client that arrives at time i by c_i. In total, we locate $3k$ clients such that for each i, the ith client c_i is positioned adjacent to client c_{i+1}. We first observe that $\text{rad}(OPT, G_{P_n}) = 1$. Algorithm A opens the first facility at c_1. Next, we locate $k/2$ *groups* of clients, where each of them contains 6 clients except the last group that has 5 clients. Note that the clients in the first group are $c_2, c_3, \cdots, c_7$, the clients in the second group are $c_8, c_9, \cdots, c_{13}$ and so on. Since Algorithm A has $k-1$ facilities left, there exists some group, say g, in which the algorithm opens at most one facility. It is easy to see that there exists a client in g whose distance to its nearest facility is 2. This completes the proof. □

Since an algorithm for the AEC model can open a facility on the location of any existing client, we can use the algorithm MRCPathAlgorithm described in Section 3.2 to solve the problem for the AEC model optimally for paths. Therefore, by Theorem 3.2.2, we have the following theorem:

Theorem 4.1.2. *There exists a 2-competitive algorithm for the AEC model of the online k-center problem for paths against both linear and strong adversaries.*

4.2 Trees and General Graphs

In this section, we present our results for trees and general graphs. We first provide a lower bound on the competitive ratio of any online algorithm against a strong adversary. Recall the lower bound $2k - 1$ on the competitive ratio of any online algorithm against a strong adversary for trees and general graphs under the MRC model; see Theorem 3.3.2. Since an algorithm can open a facility on the location of any existing client in the AEC model, a broader set of algorithmic strategies is possible, allowing for potentially improved performance. We first show a lower bound of $\frac{2(k-1)}{3}$.

Theorem 4.2.1. *There is a sequence of clients for which any online algorithm has a competitive ratio of at least $\frac{2(k-1)}{3}$ for the AEC model of the online k-center problem on trees against a strong adversary.*

Proof. The proof is similar to that of Theorem 3.3.2. The adversary constructs a tree by locating clients in a finite number of steps. Let A be any online algorithm for this problem. Algorithm A may open more than one facility in each step (note that, if Algorithm A does not open any facility for the clients that arrive in some step, then the construction stops). Consider the set of first facilities that are opened in each step; the key is that once the algorithm opens the first facility in each step, we will keep track of the radius immediately after such a facility was opened. Recall that

OPT denotes an optimal offline algorithm.

Step 1 We reveal the first sequence of clients, $\mathcal{I}_1$, which contains $3k$ clients. Algorithm A opens the first facility at the first client. There are two possibilities:

Case 1: Algorithm A does not open any facility at c_i, $2 \leq i \leq 3k$, before the arrival of c_{3k+1}. Then, we position the all remaining clients (i.e., $c_{3k+1}, c_{3k+2}, \ldots, c_n$) adjacent to client c_{3k-1}. Thus, after the arrival of c_n, we have $\text{rad}(OPT, G_{P_n}) = 1$ and $\text{rad}(A, G_{P_n}) \geq k$. Therefore, $\textsc{cr} \geq k > 2(k-1)/3$. Note that the construction stops here and does not proceed to Step 2.

Case 2: Algorithm A opens at least one facility on the location of c_i, where $2 \leq i \leq 3k$, before the adversary reveal c_{3k+1}. Then, we proceed to the next step.

Step 2 In addition to the clients opened in the previous step, the adversary reveals the next sequence of clients as $\mathcal{I}_2 = \langle \mathcal{I}_1, \ldots, \mathcal{I}_1 \rangle$, where $|\mathcal{I}_2| = k-1$. Again:

Case 1: Algorithm A does not open a facility for clients within this step. Then, the adversary locates all remaining clients adjacent to client $c_{k(3k)-1}$. Thus, after the arrival of c_n, we have $\text{rad}(OPT, G_{P_n}) \leq 3k/2$. On the other hand, $\text{rad}(A, G_{P_n}) \geq \frac{3k(k-1)}{3} = k(k-1)$ because while the adversary is locating the remaining clients, Algorithm A can open a facility for $3k(k-1)$ clients (i.e., the clients that have revealed in this step) and divide the radius by 3. Therefore, $\textsc{cr} \geq \frac{k(k-1)}{3k/2} = 2(k-1)/3$. The construction stops here and does not proceed to Step 3.

Case 2: Algorithm A opens at least one facility for clients within this step before the adversary locates client $c_{k(3k)+1}$ (i.e., the first client of the next step). Then, we proceed to the next step.

The adversary continues following these steps, until one of the following happens:

1. For some step i, $2 < i < k$, Algorithm A opens no facility for the clients that arrived in Step i, before the adversary locates the first client of Step $i+1$. Then, the adversary positions all remaining clients (i.e., the clients that are within the Steps $i+1, i+2, \ldots, k$) adjacent to the client c_j that is within Step i such that $j \geq j'$, for all clients $c_{j'}$ within Step i[1]. Let $T(i)$ denote the total number of clients that have arrived up to Step i. Then:

$$T(i) = \begin{cases} 3k, & \text{if } i = 1 \\ k \cdot T(i-1), & \text{if } i > 1 \end{cases} \tag{4.1}$$

   Since Algorithm A has not opened a facility for clients within Step i, we have $\text{rad}(A, G_{P_i}) \geq \frac{T(i-1)\cdot(k-1)}{3}$ and $\text{rad}(OPT, G_{P_i}) \leq T(i-1)/2$. Therefore, $\text{CR} \geq \frac{(k-1)\cdot T(i-1)}{3T(i-1)/2} = 2(k-1)/3$.

2. We reach Step $i = k$, where Algorithm A has opened all facilities in the previous steps. Then, the adversary locates the sequence $\mathcal{I}_k = \langle \mathcal{I}_{k-1}, \ldots, \mathcal{I}_{k-1} \rangle$, $|\mathcal{I}_k| = k-1$, of clients as a path P such that the first client on P is adjacent to client c, where c is the last client that the adversary has positioned before revealing the path P. Since Algorithm A has no more facilities and by (4.1),

[1] In other words, c_j is the last client within Step i.

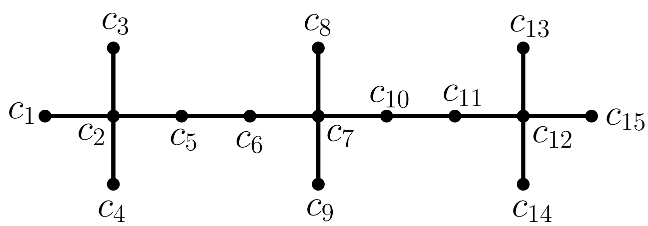

Figure 4.2: An example of the graph G in support of Theorem 4.2.3 with $k = 3$ and $n = 15$. The label of the nodes indicates the order in which the adversary reveals the corresponding clients.

we have $\text{rad}(A, G_{P_n}) \geq T(k-1) \cdot (k-1)$ and $\text{rad}(OPT, G_{P_n}) \leq T(k-1)/2$. Therefore, $\text{CR} \geq 2(k-1)/3$. This completes the proof. □

We observe that this lower bound also applies to general graphs against a strong adversary.

Since an algorithm for the AEC model can open a facility on the location of any existing client, Algorithm StrongMRC described in Section 3.3.1 is also a $(3k)$-competitive algorithm for the AEC model. Moreover, by Observation 3.3.1, this algorithm also works for trees. This gives the following theorem:

Theorem 4.2.2. *There exists a $(3k)$-competitive algorithm against a strong adversary for the AEC model of the online k-center problem for both trees and general graphs.*

For the strong adversary setting, there is a gap between the lower bound $\frac{2(k-1)}{3}$ and the upper bound $3k$ for trees and general graphs. Next, we present our results for trees and general graphs against a linear adversary.

Theorem 4.2.3. *The competitive ratio of any online algorithm for trees is at least 2 for the AEC model of the online k-center problem against a linear adversary.*

Proof. We construct a tree as follows. Consider a $3 \times 3k$ grid in which any two adjacent grid vertices have distance 1 and let the bottom-left corner of the grid be the point with coordinate (0, 0); see Figure 4.2. Let A be an online algorithm for this problem. We denote by $c_{i,j}$, $1 \leq i \leq 3$ and $1 \leq j \leq 3k$, the client positioned at the point with coordinate (i, j) on the grid. We define f_i, where $i = 0, 3, 6, \ldots$, to be the following set of clients (we call each f_i a *block*):

$$f_i = \langle c_{0+i,1}, c_{1+i,1}, c_{1+i,2}, c_{1+i,0}, c_{2+i,1} \rangle.$$

Now, we reveal the following sequence of clients to algorithm A, which contains $n = 5k$ clients

$$\mathcal{I} = (f_0, f_3, f_6, \ldots, f_{3(k-1)}).$$

We observe that OPT opens the facilities at clients $c_{1+i,1}, i = 0, 3, 6, \ldots, 3(k-1)$ in the final graph G_{P_n} and we have $\mathrm{rad}(OPT, G_{P_n}) = 1$. On the other hand, there are two cases for Algorithm A (note that A must open the first facility at $c_{0,1}$):

Case 1 A opens at least one facility at either $c_{1,1}$ or $c_{2,1}$. Then, since $\mathcal{I}$ contains exactly k blocks, we conclude that there is at least one block, say f_j, for which A has opened no facility on its clients. So, the distance between $c_{1+j,2}$ (or $c_{1+j,0}$) and its nearest facility is at least 2, and we have

$$\text{CR} = \frac{\mathrm{rad}(A, G_{P_n})}{\mathrm{rad}(OPT, G_{P_n})} \geq \frac{2}{1} = 2.$$

Case 2 A opens a facility at neither $c_{1,1}$ nor $c_{2,1}$. Thus, the distance between $c_{2,1}$ and its nearest facility is 2, and we have

$$\text{CR} = \frac{\mathrm{rad}(A, G_{P_n})}{\mathrm{rad}(OPT, G_{P_n})} \geq \frac{2}{1} = 2.$$

□

The lower bound of 2 also applies to general graphs. Recall that in the AEC model an algorithm can open a facility on the location of any existing client. This means that any online algorithm against a linear adversary for the MRC model of the online k-center problem also works for the AEC model with the same performance when the adversary remains linear. Therefore, Algorithm LinearMRC (see Section 3.3.2) is a $(2M)$-competitive algorithm, where $M = n/k$, for the AEC model. Thus, we have the following result:

Corollary 4.2.4. *There exists a* $(2M)$*-competitive algorithm, where* $M = n/k$*, for the AEC model of the online* k*-center problem for both trees and general graphs against a linear adversary.*

Chapter 5

The Online k-Median Problem: The MRC Model

In this chapter, we investigate the MRC model of the online k-median problem against both strong and linear adversaries. In each of the following sections, we consider the problem for one of the graph representations. Table 2.3 summarizes the results for the MRC model of the online k-median problem.

Throughout this chapter, we use the same notation as defined in Section 3.1. Moreover, given an online algorithm A that has opened some facilities on graph G_{P_i}, the *median* obtained by A is the sum of distances between any client to its nearest facility. We denote the median obtained by A on G_{P_i} by $\text{med}(A, G_{P_i})$.

5.1 Paths

In this section, we consider the MRC model of the online k-median problem for paths. It is straightforward that the strong and linear adversaries are similar for paths in this model. We use a linear adversary in the following result, which consequently gives the same lower bound for a strong adversary.

Theorem 5.1.1. *Given that the underlying graph G_{P_n} is a path, there is a lower bound of $\frac{2k}{2k-1}$ on the competitive ratio of any online algorithm for the MRC model of the online k-median problem against a linear adversary.*

Proof. Let A be an online algorithm and let $n = mk$, for some positive constant m. The adversary locates clients on the plane sequentially from left to right. Now, consider each m consecutive clients in the sequence. Since Algorithm A opens the first facility on the location of the first client, we have

$$\begin{aligned}
\mathrm{med}(A, G_{P_n}) &\geq (2k-1)\sum_{i=1}^{\frac{n}{2k-1}} i \\
&= (2k-1)\left(\frac{\frac{n}{2k-1}(\frac{n}{2k-1}+1)}{2}\right) \\
&= (2k-1)\left(\frac{n(n+2k-1)}{2(2k-1)^2}\right) = \frac{n(n+2k-1)}{2(2k-1)}.
\end{aligned}$$

Furthermore,

$$\mathrm{med}(OPT, G_{P_n}) \leq 2k\sum_{i=1}^{\frac{n}{2k}} i = 2k\left(\frac{\frac{n}{2k}(\frac{n}{2k}+1)}{2}\right) = k\left(\frac{n(n+2k)}{4k^2}\right) = \frac{n(n+2k)}{4k}.$$

Therefore,

$$
\begin{aligned}
\textsc{cr} &= \frac{\mathrm{med}(A, G_{P_n})}{\mathrm{med}(OPT, G_{P_n})} \\
&\leq \left(\frac{n(n+2k-1)}{4k-2}\right) / \left(\frac{n(n+2k)}{4k}\right) \\
&= \left(\frac{n+2k-1}{2k-1}\right) / \left(\frac{n+2k}{2k}\right) \leq \left(\frac{n+2k}{2k-1}\right) / \left(\frac{n+2k}{2k}\right) \\
&= \frac{2k}{2k-1}.
\end{aligned}
$$

□

Next, we present a $(1+\frac{1}{k})$-competitive algorithm that we call Algorithm PATH-MEDIAN.

Algorithm. The algorithm opens the first facility on the location of the first client. Moreover, on the arrival of client c_i, $i \geq 2$, if $\mathrm{rad}(\textsc{PathMedian}, G_{P_i}) > n/k$, then the algorithm opens a new facility at c_i.

Analysis. After the arrival of every client, the diameter of the graph increases by exactly one. Since the algorithm opens a facility whenever the radius of the graph is greater than n/k, it will never run out of facilities. Moreover, this implies that the algorithm will eventually open all the facilities.

Let c_i and c_j be two clients on which respective facilities are open and the algorithm has not opened a facility for any client on the path from c_i to c_j. We observe that $\mathrm{dist}(c_i, c_j) \leq n/k$. Therefore, for every client c' located on the path from c_i to c_j, $\mathrm{dist}(c', f) \leq n/2k$, where f is the closest facility to c'.

Let f_k be the last facility. The number of remaining clients after opening f_k is at most n/k since, otherwise, we conclude that $diam(G_{P_n}) > n-1$, which is a contradiction. Therefore, (to simplify the calculations, let $m = n/k$)

$$\begin{aligned}
\mathrm{med}(\textsc{PathMedian}, G_{P_n}) &\leq 2(k-1)\sum_{i=1}^{\frac{m}{2}} i + \sum_{i=1}^{m} i \\
&= 2(k-1)\left(\frac{\frac{m}{2}(\frac{m+2}{2})}{2}\right) + \frac{m(m+1)}{2} \\
&= (k-1)\left(\frac{m(m+2)}{4}\right) + \frac{m(m+1)}{2} \\
&= \frac{mk(m+2)}{4} - \frac{m(m+2)}{4} + \frac{2m(m+1)}{4} \\
&\leq \frac{mk(m+2)}{4} + \frac{m(m+2)}{4}.
\end{aligned}$$

On the other hand,

$$\mathrm{med}(OPT, G_{P_n}) \geq 2k\sum_{i=1}^{\frac{m}{2}} i = 2k\left(\frac{\frac{m}{2}(\frac{m+2}{2})}{2}\right) = k\left(\frac{m(m+2)}{4}\right).$$

Therefore, we have $\textsc{cr} \leq 1 + \frac{1}{k}$.

5.2 Trees and General Graphs

In this section, we describe our results for trees and general graphs. Note that, Observation 3.3.1 also applies for the online k-median problem. We first give the results for the strong adversary setting. Then we consider the problem against a linear adversary in Section 5.2.2.

5.2.1 Strong Adversary

In this section, we examine the online k-median problem against a strong adversary. The following theorem provides a lower bound of k on the competitive ratio of any online algorithm for this problem.

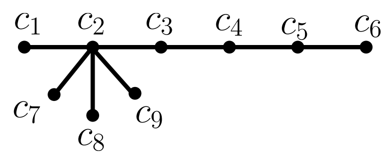

Figure 5.1: An example of graph G in support of Theorem 5.2.1 with $k = 2$ and $n = 9$.

Theorem 5.2.1. *For any online algorithm $\mathcal{A}$ for the MRC model of the online k-median problem, there exists some sequence of clients for which the competitive ratio of $\mathcal{A}$ is at least k against a strong adversary.*

Proof. Given a fixed value of k and any online algorithm A, we construct a tree G incrementally as follows. We divide locating clients into at most k steps. Let $\mathcal{I}_i$, $1 \leq i \leq k$, denote the sequence of clients that are revealed in Step i. We continue locating the sequences until after some sequence $\mathcal{I}_i$, where $1 \leq i \leq k$, we have a tree on which the median obtained by Algorithm A is at least k times that of an optimal offline algorithm. Note that the construction then stops and does not proceed to Step $i + 1$. In the following we describe the settings.

Step 1. We locate $\mathcal{I}_1$, which contains $3k$ clients, such that any two consecutive clients are adjacent, see Figure 5.1. There are two possibilities:

Case 1 Algorithm A does not open a facility for client c_i, where $2 \leq i \leq c_{3k}$. Now, we locate the remaining clients all adjacent to c_2 (see Figure 5.1). Therefore, we have $\text{med}(OPT, G_{P_n}) \leq 2k$ and $\text{med}(A, G_{P_n}) \geq 3k(3k+1)/2$. Thus, CR $\geq k$. The construction stops here and does not proceed to Step 2.

Case 2 Algorithm A opens a facility for some client c_{t_1}, where $2 \leq t_1 \leq 3k$.

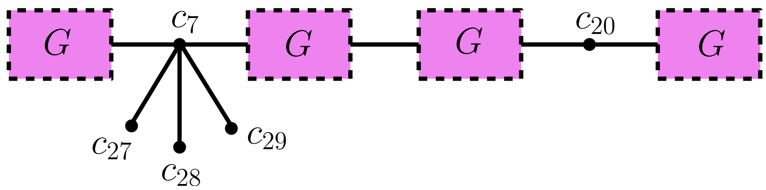

Figure 5.2: An example of graph H with $k = 2$ and $n = 29$.

Then, we proceed to Step 2.

Step 2. In this step, we reveal the following sequence of clients in addition to the clients in Step 1:

$$\mathcal{I}_2 = \langle \mathcal{I}_1, \mathcal{I}_1, \ldots, \mathcal{I}_1 \rangle,$$

where $\mathcal{I}_2$ contains $2k-1$ instances of sequence $\mathcal{I}_1$. Consider the graph H in Figure 5.2. We show each instance of graph G by a square, called *block*. Each block contains exactly $3k$ clients. Specifically, the first block is the instance of graph G that was constructed in Step 1. Note that, we open one more client after the arrival of the last client in block i, where $i \mod 2 = 1$. There are two cases for Algorithm A:

Case 1 Algorithm A does not open a facility on some client c_{t_2}, where $3k+1 \leq t_2 \leq 2k(3k)$. Then, we will locate all remaining clients adjacent to client c_{3k+1} (see Figure 5.2), and we have

$$\mathrm{med}(A, G_{P_n}) \geq \sum_{i=1}^{3k(2k-1)} i = \frac{3k(2k-1)[3k(2k-1)+1]}{2},$$

and,

$$\mathrm{med}(OPT, G_{P_n}) \leq 2k \sum_{i=1}^{3k} i = 2k \left(\frac{3k(3k+1)}{2} \right).$$

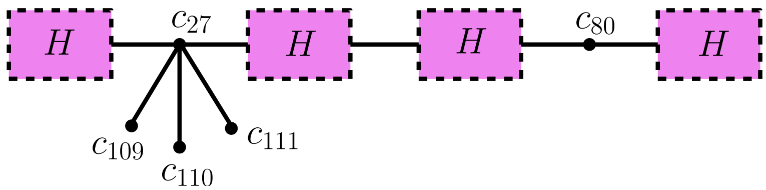

Figure 5.3: An example of graph I with $k = 2$ and $n = 111$.

Therefore, CR $\geq k$. Note that, the construction stops and does not proceed to the next step.

Case 2 Algorithm A opens a facility on at least one client located in this step, i.e., some client c_{t_2}, where $3k + 1 \leq t_2 \leq (2k)3k$. Then, we proceed to the next step.

Step 3. In this step, we reveal the following sequence of clients in addition to the clients located in the previous steps:

$$\mathcal{I}_3 = \langle \mathcal{I}_2, \mathcal{I}_2, \ldots, \mathcal{I}_2 \rangle,$$

where $\mathcal{I}_3$ contains $2k - 1$ instances of the input sequence $\mathcal{I}_2$. In other words, we have $2k - 1$ copies of graph H instead of graph G in this step. See Figure 5.3. We show each instance of graph H by a square, namely a *block*. There is a client (which does not belong to any block) after the arrival of the last client in block i, where $i \mod 2 = 1$. There are two possibilities for Algorithm A to decide:

Case 1 Algorithm A does not open a facility for some client c_{t_3}, where $6k^2+1 \leq t_3 \leq (2k)[6k^2]$. Then, we locate all remaining clients adjacent to client $c_{(2k)3k+k+1}$, and we have (see Figure 5.3),

$$\text{med}(A, G_{P_n}) \geq \sum_{i=1}^{6k^2(2k-1)} i = \frac{6k^2(2k-1)[6k^2(2k-1)+1]}{2} \in \Omega(k^6).$$

On the other hand,

$$\text{med}(OPT, G_{P_n}) \leq 2k \sum_{i=1}^{2k(3k)} i = 2k\left(\frac{2k(3k)[6k^2+1]}{2}\right) \in O(k^5).$$

Therefore, CR $\in \Omega(k)$. We stop the construction and do not proceed to the next step.

Case 2 Algorithm A opens a facility on at least one of the clients that was located in this step. Then, we proceed to the next step.

We continue steps in a similar way until one of the followings happen:

1. For some Step i, where $3 < i < k$, Algorithm A does not open a facility on the location of any clients that is revealed during Step i. To compute CR, we introduce the following recurrence relation. Let $T(i)$ denote the total number of clients at the end of Step i. Then,

$$T(i) = \begin{cases} 3k, & \text{if } i = 1 \\ 2k \cdot T(i-1), & \text{if } i > 1 \end{cases} \tag{5.1}$$

   By solving (5.1), we obtain the following closed form:

$$T(i) = 3k(2k)^{i-1}, i \geq 1. \tag{5.2}$$

Since A has opened no facility for clients in Step $i > 3$,

$$\begin{aligned}
\operatorname{med}(A, G_{P_n}) &\geq \sum_{j=1}^{T(i-1)(2k-1)} j \qquad (5.3)\\
&= \frac{T(i-1)\cdot(2k-1)[T(i-1)(2k-1)+1]}{2}\\
&= \frac{3k(2k)^{i-2}\cdot(2k-1)[3k(2k)^{i-2}(2k-1)+1]}{2}, \qquad \text{by (5.2)}\\
&\in \Omega(k^{2i}),
\end{aligned}$$

while

$$\begin{aligned}
\operatorname{med}(OPT, G_{P_n}) &\leq 2k \sum_{j=1}^{T(i-1)} j \qquad (5.4)\\
&= 2k\left(\frac{T(i-1)(T(i-1)+1)}{2}\right)\\
&= 2k\left(\frac{3k(2k)^{i-2}[3k(2k)^{i-2}+1]}{2}\right), \qquad \text{by (5.2).}\\
&\in O(k^{2i-1}).
\end{aligned}$$

Therefore, $\textsc{cr} \in \Omega(k)$.

2. We found no such i in Part 1. Then, we reach Step $i = k$, where Algorithm A has no facility to open. We reveal the sequence $\mathcal{I}_k = \langle \mathcal{I}_{k-1}, \mathcal{I}_{k-1}, \ldots, \mathcal{I}_{k-1}\rangle$, where $|\mathcal{I}_k| = 2k-1$. By setting i to k in (5.3) and in (5.4), we conclude that $\operatorname{med}(A, G_{P_n}) \in \Omega(k^{2k})$ and $\operatorname{med}(OPT, G_{P_n}) \in O(k^{2k-1})$. Therefore, $\textsc{cr} \in \Omega(k)$. This completes the proof.

□

Remark 5.2.2. *The lower bound of $\Omega(k)$ also applies to general graphs. It seems challenging to find an algorithm with competitive ratio $O(k)$ for this problem, matching* *the lower bound of $\Omega(k)$. Designing such an algorithm is an interesting direction for future work.*

5.2.2 Linear Adversary

In this section, we consider the MRC model of the online k-median problem for trees and general graphs against a linear adversary. We first give a lower bound of $\sqrt{m}$, where $m = n/k$, on the competitive ratio of any online algorithm.

Theorem 5.2.3. *For any online algorithm A for the MRC model of the online k-median problem and against linear adversary, there exists a tree on which the competitive ratio of A is at least $\sqrt{m}$, where $m = n/k$.*

Proof. The proof proceeds by an adversarial strategy. We construct a finite set of trees until we obtain a tree for which any online algorithm A has competitive ratio of at least $\sqrt{m}$ on that tree.

Let $\mathcal{C}$ denote the set of trees, initially empty. Let $\mathcal{T}_i$ denote the tree that consists of clients $c_1, c_2, \ldots, c_i$. If Algorithm A opens a facility on the location of client c_i, then $\mathcal{T}_{i-1}$ is added to $\mathcal{C}$. The clients are located as follows.

The adversary first locates m/c clients in the plane as a path, for some constant $c > 0$ whose value will be specified later. Then, at most $(\frac{c-1}{c})m$ more clients are located all adjacent to the last client of the sequence (i.e., client $c_{m/c}$). We call this set of m clients a *plant* with c_1 as the current *root* of the plant. Moreover, we call the first m/c clients on the path a *stem* and the clients that are adjacent to $c_{m/c}$ a *flower*, see Figure 5.4(a) for an illustration. Next, the adversary starts from client c_1 and constructs another similar plant in some other direction. We observe that at most

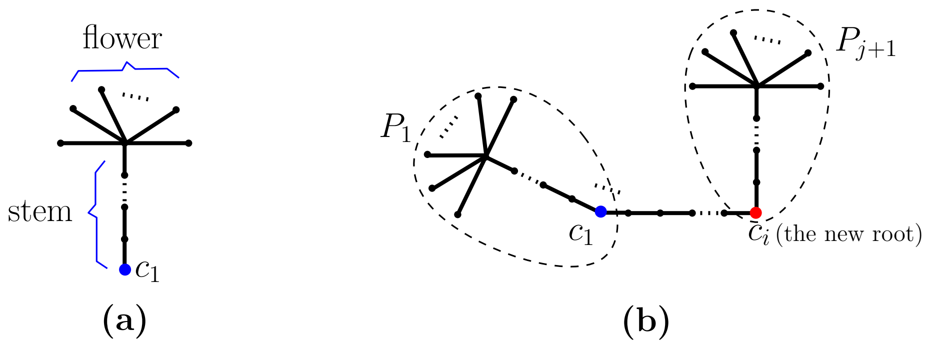

Figure 5.4: (a) An example of a plant. (b) The algorithm opens a facility on a client located on the stem of plant P_j.

k plants are constructed. The construction of plants is continued until Algorithm A opens a facility on the location of some client c_i. Let P_j denote the jth plant, $1 \leq j \leq k$, and suppose that c_i is located in plant P_j.

- If c_i is located on the stem of P_j, then (i) we add $\mathcal{T}_i$ to $\mathcal{C}$ and, (ii) we consider c_i, instead of c_1, as the root for the next plants, see Figure 5.4(b) for an illustration.
- If c_i is located on the flower of P_j, then (i) we add $\mathcal{T}_{i-1}$ to $\mathcal{C}$ and, (ii) we leave completing the current plant and start constructing a new plant from the current root.

At the end, we add $\mathcal{T}_n$ to $\mathcal{C}$ as the last tree. It is not hard to see that Algorithm A opens the facilities only on the location of the roots of $\mathcal{T}_n$. If Algorithm A opens at most one facility, then the theorem follows. Thus, we assume that Algorithm A opens at least two facilities. Let $\mathrm{dist}_A(f_i, f_j)$ denote the distance between two facilities f_i and f_j opened by Algorithm A. We first show the following result.

Lemma 5.2.4. *Given two algorithms A_1 and A_2 running on $\mathcal{T}_n$, if $\min_{i,j} \mathrm{dist}_{A_1}(f_i, f_j) < m/c$ and $\min_{i,j} \mathrm{dist}_{A_2}(f_i, f_j) \geq m/c$, then the median obtained by A_1 is greater than* *the median obtained by* A_2*.*

Proof. Based on the construction of $\mathcal{T}_n$ described above, whenever an algorithm opens a facility the adversary updates the root of the plant. In other words, the algorithm opens facilities only on the roots. Moreover, we observe that the distance between any two facilities opened by the algorithm cannot be greater than m/c because the length of the stem of a plant is m/c.

Since there exists two facilities opened by A_1 with distance less than m/c, the total number of clients on the flowers of plants for A_1 is greater than that for A_2. Therefore, the median obtained by A_1 is greater than the median obtained by A_2. $\square$

By Lemma 5.2.4, we conclude that the best *online* algorithm, say A^*, run on $\mathcal{T}_i$ is one with $\mathrm{dist}(f_i, f_j) = m/c$ for every two opened facilities f_i and f_j. Therefore, Algorithm A^* opens the facilities on the roots of $\mathcal{T}_n$ such that every two roots have distance m/c, see Figure 5.5. Therefore,

$$\begin{aligned}
\mathrm{med}(A^*, G_{P_n}) &\geq k\left(\sum_{i=1}^{\frac{m}{c}} i + \sum_{i=1}^{\frac{m}{2c}} i + \left[(\frac{c-1}{c})m - \frac{m}{2c}\right]\frac{m}{c}\right) \qquad (5.5)\\
&= k\left(\frac{m}{c}\left[\frac{m+c}{c}\right]/2 + \frac{m}{2c}\left[\frac{m+2c}{2c}\right]/2 + m^2(c-1)/c^2 - m^2/2c^2\right)\\
&= k\left(\frac{4m(m+c) + m(m+2c) + 8m^2c - 12m^2}{8c^2}\right)\\
&= mk\left(\frac{8mc + 6c - 7m}{8c^2}\right).
\end{aligned}$$

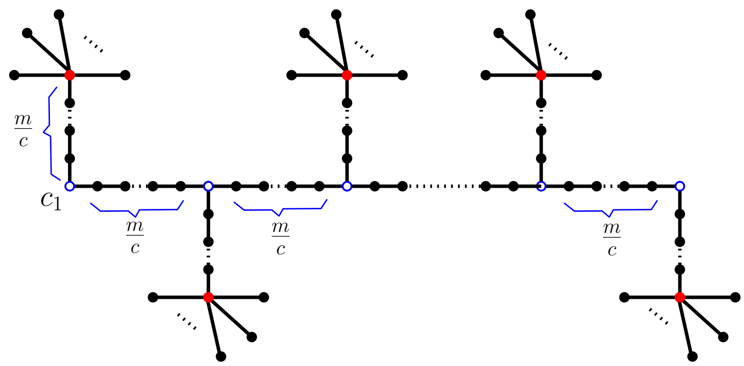

Figure 5.5: The facilities opened by the best online algorithm are shown by blue nodes. An optimal offline algorithm opens the facilities on the location of the red nodes.

Furthermore, (see Figure 5.5),

$$\begin{aligned} \mathrm{med}(OPT, G_{P_n}) &\leq k\left(\sum_{i=1}^{\frac{2m}{c}} i + (\frac{c-1}{c})m - \frac{m}{c}\right) \qquad (5.6)\\ &= k\left(\frac{2m}{c}\left[\frac{2m+c}{c}\right]/2 + m(c-1)/c - m/c\right)\\ &= k\left(\frac{4m^2 + 2mc + 2mc^2 - 2mc - 2mc}{2c^2}\right)\\ &= mk\left(\frac{2m + c^2 - c}{c^2}\right). \end{aligned}$$

By setting $c = \sqrt{m}$, we get $\textsc{cr} \in \Omega(\sqrt{m})$. Since Algorithm A^* is the best possible online algorithm, we conclude that the competitive ratio obtained by Algorithm A on $\mathcal{T}_n$ is $\Omega(\sqrt{m})$. This complete the proof. $\square$

Next, we present an $O(nm)$-competitive algorithm, where $m = n/k$, for the MRC model of the online k-median problem against a linear adversary. In fact, we show that Algorithm LinearMRC described in Section 3.3.2 (denoted by A_3) is an $O(nm)$-competitive algorithm for the online k-median problem against a linear adversary.

Analysis of Algorithm LinearMRC. If $k = 1$, then the problem becomes trivial as any online algorithm has to open the facility on the location of the first client and, therefore, $\text{CR} \le 2$. Moreover, if $n \le k$, then the algorithm opens a facility on the location of every client that is optimal and we have $\text{CR} = 1$. Therefore, in the following we assume that $k \ge 2$ and $n > k$.

Recall from Section 3.3.2 that the radius obtained by the algorithm is at most $n/(k-1)$ and $\text{rad}(OPT, G_{P_n}) \ge (\text{diam}(G_{P_n}) - k)/2k$. Thus, we know that the distance between any client to its nearest facility opened by Algorithm A_3 is at most $n/(k-1)$ and, therefore, $\text{med}(A_3, G_{P_n}) \le n^2/(k-1)$. Moreover, since an algorithm (offline or online) can open at most k facilities, there are at least $n-k$ clients in graph G_{P_n} whose distances to their nearest facility opened by OPT is at least 1. Thus, $\text{med}(OPT, G_{P_n}) \ge n-k$. Therefore,

$$\text{CR} = \frac{\text{med}(A_3, G_{P_n})}{\text{med}(OPT, G_{P_n})} \le \frac{n^2}{(k-1)(n-k)}.$$

Case 1. If $n \ge 2k$, then

$$\text{CR} = \frac{\text{med}(A_3, G_{P_n})}{\text{med}(OPT, G_{P_n})} \le \frac{n^2}{(k-1)(n-k)} \le \frac{n^2}{(k-1)(2k-k)} = \frac{n^2}{k^2-k} \le \frac{n^2}{k} = nm.$$

Case 2. If $k < n < 2k$, then $\lceil n/(k-1) \rceil = 2$ and, hence, $\text{CR} \le n$.

Therefore, we have the following result:

Theorem 5.2.5. *There exists an $O(nm)$-competitive algorithm against a linear adversary for the MRC model of the online k-median problem for both trees and general graphs.*

Chapter 6

Conclusion and Future Work

In this thesis, we introduced two online models, namely the MRC and the AEC models, for two facility location problems, namely the k-center and the k-median problems. In the MRC model, a facility can be opened only on the location of the newly-arrived client while in the AEC model, the facility can be opened on the location of any existing client. We studied these models under two types of adversaries depending on the number of clients available to the adversary; the number of clients available to a *linear* adversary is linear in the number of facilities while a strong adversary can open as many clients as it wishes.

In Chapter 3, we considered the online k-center problem under the MRC model. We proved a lower bound of 2 on the competitive ratio of any online algorithm against a linear adversary and gave a $(2n/k)$-competitive algorithm for this problem. It remains open whether there exists an α-competitive algorithm for this problem, where $\alpha \in [2, 2n/k)$. In case of a strong adversary, we showed a lower bound of $2k-1$ on the competitive ratio, where k is the number of facilities, for trees and general graphs. We also gave a $3k$-competitive algorithm for this problem. The question of finding an α-competitive algorithm, where $\alpha \in [2k-1, 3k)$, remains open for future work.

In Chapter 4, we studied the online k-center problem for the AEC model. We proved a lower bounds of 2 on the competitive ratio of any online algorithm against a linear adversary. We then showed a lower bound of $2(k-1)/3$ on the competitive ratio of any online algorithm against a strong adversary. Our algorithms for the AEC model were adopted from the MRC model. Since an algorithm for the AEC model is more powerful than an algorithm for the MRC model, algorithms with better competitive ratios may be possible for the AEC model. Giving algorithms with better competitive ratios is another direction for future work.

We investigated the online k-median problem under the AEC model in Chapter 5. When the underlying graph is constrained to a path, we showed a lower bound of $2k/(2k-1)$ for both linear and strong adversaries. We also gave an algorithm with competitive ratio of $1+1/k$ for this problem. In case of trees and general graphs, we presented an $O(nm)$-competitive algorithm against a linear adversary. Moreover, we showed a lower bound of $\Omega(1)$ for this problem. For strong adversaries, we showed a lower bound of $\Omega(k)$ on the competitive ratio of any online algorithm for the online k-median problem under the MRC model. The question of designing an online algorithm with competitive ratio of $O(k)$, matching the lower bound of $\Omega(k)$, remains open.

We did not study the online k-median problem under the AEC model in this thesis. As in the offline version of the k-median problem, solving the online k-median problem under this model seems more challenging. Exploring the online k-median problem under the AEC model provides another direction for future work.

Bibliography

[1] Pankaj Agarwal and Micha Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30:412–458, 1999.

[2] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In *Proceedings of the Symposium on Theory of Computing (STOC '98), Dallas, TX, USA*, pages 106–113, 1998.

[3] Sergei Bespamyatnikh, Binay Bhattacharya, Mark Keil, David Kirkpatrick, and Michael Segal. Efficient algorithms for centers and medians in interval and circular-arc graphs. *Networks*, 39:144–152, 2002.

[4] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[5] Prosenjit Bose, Anil Maheshwari, and Pat Morin. Fast approximations for sums of distances, clustering and the fermat-weber problem. *Computational Geometry: Theory and Applications (CGTA)*, 24:135–146, 2003.

[6] Timothy Chan. More planar two-center algorithms. *Computational Geometry: Theory and Applications (CGTA)*, 13(3):189–198, 1999.

[7] Ramaswamy Chandrasekaran and Arie Tamir. Algebraic optimization: the fermat-weber location problem. *Mathematical Programming*, 46:219–224, 1990.

[8] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS '99), New York City, NY, USA*, pages 378–388, 1999.

[9] T. C. Edwin Cheng, Liying Kang, and C. T. Ng. An improved algorithm for the p-center problem on interval graphs with unit lengths. *Computers & Operations Research*, 34(8):2215–2222, 2007.

[10] Reza Dorrigiv. *Alternative measures for the analysis of online algorithms*. PhD thesis, University of Waterloo, 2010.

[11] Stephane Durocher, Krishnam Raju Jampani, Anna Lubiw, and Lata Narayanan. Modeling gateway placement in wireless networks: Geometric k-centers of unit disc graphs. *Computational Geometry: Theory and Applications (CGTA)*, 44(5):286–302, 2011.

[12] Tomas Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th the Symposium on Theory of Computing (STOC '88), Chicago, IL, USA*, pages 434–444, 1988.

[13] Greg Frederickson. Parametric search and locating supply centers in trees. In *Proceedings of Workshop on Algorithms and Data Structures (WADS '91), Ottawa, ON, Canada*, pages 299–319, 1991.

[14] Teofilo Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[15] S. Louis Hakimi. Location theory. *Rosen, Michaels, Gross, Grossman, and Shier, editors, Handbook of Discrete and Combinatorial Mathematics*, 2000.

[16] Wen-Lian Hsu and George Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[17] Piotr Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the Symposium on the Theory of Computing (STOC '99), Atlanta, GA, USA*, pages 428–434, 1999.

[18] Kamal Jain and Vijay V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS '99), New York City, NY, USA*, pages 2–13, 1999.

[19] Oded Kariv and S. Louis Hakimi. An algorithmic approach to network location problems I: The p-centers. *SIAM Journal of Applied Mathematics*, 37(3):513–538, 1979.

[20] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean kappa-median problem. In *Annual European Symposium on Algorithms (ESA '99)*, pages 378–389, 1999.

[21] Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David Williamson. A general approach for incremental approximation and hierarchical

clustering. In *In Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '06), Miami, FL, USA*, pages 1147–1156, 2006.

[22] Nimrod Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM Journal on Computing*, 4:759–776, 1983.

[23] Nimrod Megiddo and Kenneth Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

[24] Alexa Sharp. *Incremental algorithms: solving problems in a changing world.* PhD thesis, Cornell University, 2007.