# ONLINE PURSUIT ALGORITHMS AND OPTIMAL STRATEGIES FOR HETEROGENEOUS ROBOTS

By

Shivam Bajaj

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical and Computer Engineering—Doctor of Philosophy

2023

## ABSTRACT

The advancement in technology, especially Unmanned Aerial Vehicles (UAVs) or drones, has helped mankind in many aspects of everyday life, such as environmental monitoring and in surveillance. However, an easy access to UAV technology has spurred its malicious use, leading to numerous attempts of flying UAVs into restricted areas or in public places. One possible way to counter against such adversarially intruding UAVs is to tag or disable them before they reach a specified location by using superior drones. But the problem of how to plan the motion of these drones, i.e., designing algorithms that have provable guarantees on the numbers of adversarial UAVs that can be disabled has remained an open problem.

This dissertation addresses design of control strategies and online algorithms, i.e., algorithms that do not have information about the intruders a priori, for drones to pursue and disable one or many intruders and is divided into two parts. The first part involves many, possibly infinite, intruders that move directly towards a region of interest. For this scenario, we design decentralized as well as cooperative online algorithms with provable worst-case guarantees for 1) a single drone defender, 2) a team of homogeneous defenders, and 3) a team of heterogeneous defenders. The aim of such defender drones is to capture as many intruders as possible that arrive in the environment. To quantify how well the algorithms perform in the worst-case, we adopt a competitive analysis technique. In particular, the algorithms designed in this dissertation exhibit a finite competitive ratio, meaning that the performance of an online algorithm is no worse than a finite value determined in this dissertation. We also determine fundamental limits on the existence of online algorithms with finite competitive ratios.

In terms of heterogeneity, the first part addresses drones of different capabilities as well as motion models, such as a drone and a turret operating in the same environment. The second part of this dissertation considers coupling between the motion. Specifically, this part considers a laser attached to a fixed wing aircraft. The aircraft is modelled as a planar Dubins vehicle and the laser with a finite range can rotate clockwise or anti-clockwise. We design an optimal control strategy for both the Dubins vehicle and the laser such that a static target, located in the environment, is tagged in minimum time. By applying the minimum principle, we establish cooperative properties between the laser and the Dubins vehicle. We further establish that the shortest path must lie in a family of 13 candidate paths and characterize the solutions to all of these types. Finally, this part also addresses a scenario when the location of the perimeter is not precisely known to the adversary requiring the adversary to release *surveilling agents* to estimate the location of the perimeter. Specifically, a pursuit problem of surveilling agents is considered wherein the objective of the pursuer is to capture any one of the surveilling agents whereas the surveilling agents aim to jointly maximize a weighted combination of the determinant of the Fisher Information Matrix and the distance to the pursuer from the nearest tracker. We establish the optimal strategy for the pursuer and show that the optimization problem for the surveilling agents can be converted to a Quadratically Constrained Quadratic Program. We further establish that the optimal strategies obtained for the pursuer and the trackers form a Nash equilibrium of this game.

Dedicated to my family.

**ACKNOWLEDGEMENTS**

I would like to express my deepest appreciation to my advisor, Dr. Shaunak D. Bopardikar. He has not only taught me research but has also provided unwavering support and constructive criticism throughout my research projects. I would also like to thank my committee members, Dr. Eric Torng, Dr. Xiaobo Tan, and Dr. Vaibhav Srivastava for their time, support, and valuable feedback.

I would also like to express my gratitude towards my collaborators, Dr. Alexander Von Moll, Dr. Isaac Weintraub, Dr. Eloy Garcia, and Dr. David Casbeer at the Control Science Center in Air Force Research Laboratory for numerous insightful discussions. Special thanks to Dr. Bhargav Jha for a wonderful joint work.

I am very grateful to all my lab mates and friends, especially Sandeep Banik, for all the discussions over coffee breaks as well as the treasured memories. I would also like to thank Dr. Shardula Gawankar for continuously motivating me as well as supporting me throughout this journey. Finally and most importantly, I would like to thank my family, without whom none of this would be possible, for their immense support, love, sacrifice, and continuously motivating me all this while.

## TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

The advancement in technology, especially Unmanned Aerial Vehicles (UAVs) or drones, has helped mankind in many aspects of everyday life, such as monitoring [43, 21] and surveillance [53, 67]. However, the malicious use of such technology has also increased; recent instances include flying UAVs into restricted areas or crash landing [71] and using drones for surveillance in an ongoing armed conflict [42]. Although these instances use a single drone, a scenario involving hundreds or thousands of such drones is also feasible. An excerpt from [29] substantiates this statement.

"*Swarms of low-cost, autonomous air and space systems can provide adaptability, rapid upgradability, and the* ***capacity to absorb losses*** *that manned systems cannot. By leveraging advances in artificial intelligence, low-cost sensors, and networked communications, low-end systems can restore the agility to* ***attack adversary weaknesses*** *in unexpected ways by* ***exploiting numbers*** *and complexity.*"

Simply put, by using many robotic drones (referred to as *intruders* from here on), or equivalently a robotic swarm, an adversary can severely damage critical infrastructures. To counter such intruders, superior drones (referred to as *defenders* from this point onward) may be utilized for defense applications, including patrolling, surveillance, or perimeter defense. Deploying these defenders for such applications requires designing motion plans with certain provable guarantees. However, as expected, the information possessed by defenders about the intruders or vice-versa plays a crucial role in designing the motion plans for the defenders.

This dissertation sheds light on perimeter defense and surveillance scenarios under information asymmetry. Specifically, this work is in two parts. First, we consider a perimeter defense scenario in which the defenders do not have complete information about the intruders and thus, must take decisions as and when new information is released. Then we consider two surveillance scenarios in which 1) the defenders are used to gain information about critical locations and 2) deter intruders from gaining critical information.

When defenders do not have complete information about the intruders, the defenders must take decisions as and when new information is released to them while ensuring certain guarantees for the application in consideration. For instance, consider a scenario in which a single intruder, A, arrives near a high value facility which is being guarded by a defender, D. If D does not attack A, by say *pursuing* A, then the facility faces a risk. So D, given the information of A, must plan its motion to pursue A.

Now, while D is in pursuit of A, consider that a swarm S of intruders arrive. In fact, it may be the case that A deliberately lured D away from the facility so that S can damage the facility. In such a scenario, was pursuing A a valid motion plan for D? Could D have done better by chasing A sufficiently away from the facility and then returning to the facility? Clearly, if the arrival of S was known a priori, one may have designed a better motion plan for D. However, in this scenario, D wasted valuable time pursuing A and, as a consequence, the facility faces a risk.

This example highlights an *online optimization problem* or an online problem. In simple words, the defender faces incomplete information about the arrival of intruders and must plan its motion as new information is released over time. Thus a critical question arises:

***Question 1 (Perimeter Defense):*** *How to design motion plans with certain guarantees for the defenders that seek to defend a perimeter against many intruders and without any information about the intruders arrival process?*

Similarly, in surveillance applications, the defenders may be used either to 1) deter intruders from gaining information (maintaining informational advantage) or 2) gather information (gaining informational advantage) leading to the following question:

***Question 2 (Surveillance):*** *How to design motion plans with certain guarantees for the defenders to maintain or gain informational advantage?*

This work addresses both Question 1 and Question 2. In particular, we address the first question by focusing on non-maneuverable intruders that seek to damage a facility.[1]

[1]We will see in Chapter 2 that some of the results can be applied to maneuverable intruders with slight modifications.

More precisely, we consider *perimeter defense problems* in which defenders seek to defend a perimeter from an arbitrary number of intruders that arrive in the environment at arbitrary time instants and locations. As the defenders do not have the information about the number, arrival locations, and arrival times of the intruders, we design online algorithms with provable guarantees as well as fundamental limits for the defenders for the worst-case scenarios. In simpler terms, irrespective of how and when the intruders arrive in the environment, the algorithms must guarantee that certain number of intruders are always captured. Further, we address the second question by focusing on designing optimal control strategies for the defenders for both, deterring intruders as well as gathering information.

We now provide some background information and review concepts that will be used in the subsequent chapters which address Question 1.

## 1.1 Competitive Analysis: A Brief Overview

Recall Question 1 that considers that the defenders do not have information about the arrival of the intruders. Such type of problems in which the information is not known a priori and is revealed over time are called online problems.

The information revealed to the defenders is modelled as a sequence, known as *input sequence* $\mathcal{I}$, which can be defined as a set of 3-tuples comprising: (i) an arbitrary time instant $t \leq T$, where $T$ denotes the final time instant, (ii) the number of intruders $N(t)$ that are released at time instant $t$, and (iii) the release location of each of the $N(t)$ intruders. Formally, $\mathcal{I} = \{t, N(t), \mathcal{L}_{N(t)}\}_{t=0}^{T}$, where $\mathcal{L}_{N(t)}$ denotes the set of locations of $N(t)$ intruders that arrive at time instant $t$. An input instance $I(t) \subseteq \mathcal{I}$ is a part of the input sequence that has been revealed until the current time instant $t$. Since the information is not known to the defenders in advance, we design *online algorithms* that only uses the information $I(t) \subseteq \mathcal{I}$ that is revealed to it until time $t$. Formally, an *online algorithm* $\mathcal{A}$ assigns velocity(ies) to the defender(s) at time $t$ as a function of the input instance $I(t) \subseteq \mathcal{I}$ revealed until time $t$.

For any online problem, we can assume the existence of an *optimal offline algorithm*, defined as a non-causal algorithm which has complete information of the entire input sequence

$\mathcal{I}$ to assign velocity(ies) to the the defender(s) at any time $t$. Such an algorithm can be used to measure the quality of an online algorithm. The idea of using an offline optimal algorithm to compare with an online algorithm was first introduced in [69] and is known as *competitive analysis*. Before we describe this technique, we briefly describe an instance of the perimeter defense problem.

A *problem instance* $\mathcal{P}$ for a perimeter defense problem (Question 1) is characterized by the following parameters:

- The speed of the intruders which is normalized by the speed of the defenders, unless otherwise stated.
- The size of the environment as well as the perimeter

Note that a problem instance $\mathcal{P}$ can also be characterized by additional properties of the defenders such as the capture radius or service times of the defenders. A more precise and formal definition of a problem instance is provided in each of the subsequent chapters.

As we consider the worst-case scenarios, an online algorithm $\mathcal{A}$'s performance is measured using the notion of *competitive ratio*: the ratio of the optimal offline algorithms performance and algorithm $\mathcal{A}$s performance for a worst-case input sequence for algorithm $\mathcal{A}$. An algorithm is $c$-competitive if its competitive ratio is no larger than $c$ which means its performance is guaranteed to be within a factor $c$ of the optimal, for all input sequences. More formally, the definition of competitive ratio and a $c$-competitive algorithm is as follows:

**Definition 1** (**Competitive Ratio for Deterministic Algorithms**). *Given a problem instance $\mathcal{P}$, an input sequence $\mathcal{I}$, and an online algorithm $\mathcal{A}$, let $n_{\mathcal{A}}(\mathcal{I},\mathcal{P})$ (resp. $n_{\mathcal{O}}(\mathcal{I},\mathcal{P})$) denote the performance of an online algorithm $\mathcal{A}$ (resp. optimal offline algorithm $\mathcal{O}$) on an input sequence $\mathcal{I}$. Then,*

*i) the competitive ratio of $\mathcal{A}$ on $\mathcal{I}$ is defined as $C_{\mathcal{A}}(\mathcal{I},\mathcal{P}) = \frac{n_{\mathcal{O}}(\mathcal{I},\mathcal{P})}{n_{\mathcal{A}}(\mathcal{I},\mathcal{P})} \geq 1$,*

*ii) the competitive ratio of $\mathcal{A}$ for the problem instance $\mathcal{P}$ is $c_{\mathcal{A}}(\mathcal{P}) = \sup_{\mathcal{I}} \; C_{\mathcal{A}}(\mathcal{I},\mathcal{P})$,*

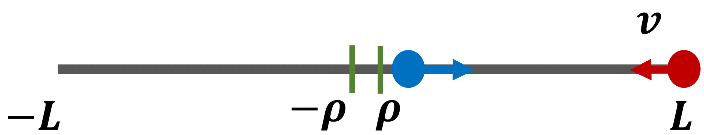



Figure 1.1 Description of Example 1. The green line segment denotes the perimeter and the blue and the red dots denotes the defender and the intruder, respectively.

*iii) the competitive ratio for the problem instance $\mathcal{P}$ is $c(\mathcal{P}) = \inf_{\mathcal{A}} c_{\mathcal{A}}(\mathcal{P})$*

*Finally, an algorithm is c-competitive for the problem instance $\mathcal{P}$ if $c_{\mathcal{A}}(\mathcal{P}) \leq c$, where $c \geq 1$ is a constant.*

In this work, the performance of an online or optimal offline algorithm is the number of intruders captured by the defenders following the respective algorithm.

Competitive analysis falls under a general framework of Request-Answer games and thus, can be viewed as a two player zero-sum game between an online player and an *adversary* [16]. An adversary is defined as a pair $(\mathcal{Q}, \mathcal{O})$, where $\mathcal{Q}$ is the input component responsible for generating the input sequences $\mathcal{I}$ and $\mathcal{O}$ is an optimal offline algorithm which maximizes $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P})$. Thus, the adversary, with the complete information of the online algorithm, constructs a worst-case input sequence so as to maximize the competitive ratio, i.e., it minimizes the number of intruders captured by the online algorithm and simultaneously maximizes the number of intruders captured by an optimal offline algorithm. On the other hand, the online player operates an online algorithm on an input sequence created by the adversary. Thus, in this work, we restrict the choice of inputs $\mathcal{I}$ to those for which there exists an optimal offline algorithm $\mathcal{O}$ such that $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P}) \geq 1$. Clearly, $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P}) \geq n_{\mathcal{A}}(\mathcal{I}, \mathcal{P})$. However, if for some $\mathcal{I}$, $n_{\mathcal{A}}(\mathcal{I}, \mathcal{P}) = 0$, then we say that $\mathcal{A}$ is not $c$-competitive for any finite $c$. We now provide an example for a better understanding of competitive ratio and the information structure. We say that a defender *captures* an intruder if it is collocated with the intruder, unless otherwise stated.

**Example 1.** *Consider a linear environment as shown in Figure 1.1 in which intruders arrive at the two endpoints $L$ and $-L$. Upon arrival, the intruders move towards the perimeter,* *defined as a set of two locations $\rho$ and $-\rho$, with a fixed speed $v$. A single defender that can move with unit speed seeks to capture maximum number of intruders before they breach the perimeter. Suppose the defender follows an algorithm which is defined as follows:*

***Simple Algorithm:*** *Move towards endpoint $+1$ and stay at the endpoint for all time.*

Recall, that an adversary has complete information of the entire algorithm a priori whereas the Simple Algorithm does not have any knowledge of how and when the intruders arrive in the environment. Given the information about Simple Algorithm, suppose that the adversary releases intruders based on the input sequence $\mathcal{I}_1$ described as follows. Release all intruders at endpoint $+1$ at time 1. Against such an input sequence, the Simple Algorithm can capture all intruders. Similarly, an optimal offline algorithm, having the complete information of $\mathcal{I}_1$, would also move its defender towards endpoint $+1$ and capture all intruders. Thus, from Definition 1(i) and against input sequence $\mathcal{I}_1$, Simple Algorithm has a competitive ratio of 1. However, this does not mean that Algorithm Simple is 1-competitive. This is explained as follows. There exists an input sequence, $\mathcal{I}_2$, which has intruders arriving only at endpoint $-1$. Against such an input sequence, Simple Algorithm does not capture any intruder. On the other hand, since an optimal offline algorithm has the information of entire $\mathcal{I}_2$ a priori, it can move its defender towards $-1$ and capture all intruders. This implies that in the worst-case, Simple Algorithm does not capture any intruder, meaning that there exists a constant greater than 1 that bounds the competitive ratio of Simple Algorithm and thus, it not 1-competitive (refer to Definition 1 (ii)). In fact, as Simple Algorithm does not capture any intruder, it is not $c$-competitive for any finite $c$.

Thus far, we have explored the concept of competitive ratio within the context of deterministic online algorithms. Now, let us turn our attention to the various adversarial models and the definition of competitive ratio for randomized online algorithms. We begin by defining a randomized online algorithm.

Let $\mathcal{A}_{det} = \{A_i : i \in \mathbb{N}\}$ denote the set of all online deterministic algorithms $A_i$ and let $n_{A,i}(\mathcal{I}, \mathcal{P})$ (resp. $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P})$) denote the performance, i.e., the number of intruders captured, of the vehicle when following a deterministic algorithm $A_i \in \mathcal{A}_{det}$ (resp. $\mathcal{O}$) on an input sequence $\mathcal{I}$. Then, a randomized algorithm $\mathcal{A}$ is defined as a probability distribution over the set of deterministic online algorithms $\mathcal{A}_{det}$.

Unlike the case of deterministic algorithms, there are two different models of the adversary; oblivious and adaptive [16]. These models differ in the information they have in generating the input sequence. In this work, we consider the oblivious adversary which is defined as follows:

**Definition 2** (**Oblivious Adversary** [16])**.** *For any online randomized algorithm, an oblivious adversary generates the entire input sequence in advance.*

Although an oblivious adversary has prior information of the randomized algorithm, it does not know the outcome of the random choices made by the randomized algorithm. Irrespective of the model of the adversary, an adversary, for a given randomized algorithm, generates input sequence such that the number of intruders captured by the optimal offline algorithm is maximized and the number of intruders captured, in expectation, by the randomized algorithm is minimized. We now define the competitive ratio for a randomized online algorithm.

Given a problem instance $\mathcal{P}$, an input sequence $\mathcal{I}$, and an optimal offline algorithm $\mathcal{O}$, the competitive ratio for randomized algorithms is defined as:

**Definition 3** (**Competitive Ratio for Randomized Algorithms**)**.** *Given a problem instance $\mathcal{P}$, an input sequence $\mathcal{I}$, and a randomized online algorithm $\mathcal{A}$, let $n_{\mathcal{A}}(\mathcal{I}, \mathcal{P})$ (resp. $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P})$) denote the performance of a randomized online algorithm $\mathcal{A}$ (resp. optimal offline algorithm $\mathcal{O}$) on an input sequence $\mathcal{I}$. Then,*

*i) the competitive ratio of $\mathcal{A}$ on $\mathcal{I}$ is defined as $C_{\mathcal{A}}(\mathcal{I}, \mathcal{P}) = \frac{n_{\mathcal{O}}(\mathcal{I}, \mathcal{P})}{\mathbb{E}[n_{\mathcal{A}}(\mathcal{I}, \mathcal{P})]} \geq 1$, where the expectation $\mathbb{E}[.]$ is with respect to the distribution over $\mathcal{A}_{det}$.*

*ii) The competitive ratio of $\mathcal{A}$ for the problem instance $\mathcal{P}$ is $c_{\mathcal{A}}(\mathcal{P}) = \sup_{\mathcal{I}} \ C_{\mathcal{A}}(\mathcal{I}, \mathcal{P})$,* *iii) the competitive ratio for the problem instance $\mathcal{P}$ is $c(\mathcal{P}) = \inf_{\mathcal{A}} c_{\mathcal{A}}(\mathcal{P})$*

*Finally, an algorithm is c-competitive for the problem instance $\mathcal{P}$ if $c_{\mathcal{A}}(\mathcal{P}) \leq c$, where $c \geq 1$ is a constant.*

We now summarize relevant prior works on perimeter defense problems.

## 1.2 Perimeter Defense and Dynamic Vehicle Routing: A Brief History

We start with the existing and relevant literature on perimeter defense problems followed by the relevant literature on dynamic vehicle routing problems.

### 1.2.1 Perimeter Defense Problems

Introduced in [41] as a target guarding problem, perimeter defense problem is a variant of pursuit evasion problems in which the aim is to determine optimal policies for the pursuers (or defenders) and evaders (or intruders) by formulating it as a differential game. Since then, several extensions have been studied such as Target-Attacker-Defender (TAD) games [47, 28] or reach-avoid games[34]. Versions of the same with multiple vehicles and intruders have been studied extensively as [24, 81, 82, 48, 36, 35]. The classic approach either requires computing solutions to the Hamilton-Jacobi-Bellman-Isaacs equation which, due to the curse of dimensionality, is applicable only for low dimensional state spaces and simple environments [51] or determining the value function of the game. Another work [46] addresses a class of perimeter defense problems, called perimeter defense games, which require the defenders to be constrained on the perimeter. We refer the reader to [68] for a review of perimeter defense games. More recently, [62, 63] considered a target guarding problem in which targets arrive sequentially according to some random process.

Considering a turret, which is a different type of defender compared to a mobile defender, [1] introduced a differential game between a turret and a mobile intruder with an instantaneous cost based on the angular separation between the two. A similar problem with possibility of retreat was considered in [77, 78]. Further, [79] and [80] considered a scenario in which the turret seeks to align its angle to that of the intruders in order to neutralize an attacker. Other recent works include approach based on control barrier functions [37] and task assignment problems [75, 76]. All of these works require that some information, such as location, of the intruders is known a priori.

### 1.2.2 Dynamic Vehicle Routing Problems

Online problems which require that the route of the vehicle be re-planned as information is revealed gradually over time are known as dynamic vehicle routing problems [64, 12, 20]. The most relevant works in this area are TRP-TW problems in which most of the works consider either zero or stochastic service times [56, 31, 59, 11, 38]. In these problems, the input (also known as demands) is static and therefore, the problem is to find the shortest route through the demands in order to minimize (maximize) the cost (reward). Examples of such metrics would be the total service time or the number of inputs serviced. In perimeter defense scenarios, the input (intruders) are not static. Instead, they are moving towards a specified region, making this problem more challenging than the former. With the assumption that the arrival process of the intruders is stochastic, [3, 70, 49] consider the perimeter defense problem as a vehicle routing problem. Other related works that do not make any assumptions on the intruders and focus on design of *must-win* algorithms, i.e., algorithms that detect every intruder in an environment, are [30, 54].

Most prior work on perimeter defense has either focused on determining an optimal strategy for few intruders or intruders generated by a stochastic process. The optimal strategy approaches do not scale with an arbitrary number of intruders released online. While stochastic approaches yield important insights into the average-case performance of defense strategies, they do not account for the worst-case in which intruders may coordinate their arrival to overcome the defense.

## 1.3 Organization

The remainder of this dissertation is organized as follows. Chapter 2 and Chapter 3 considers a perimeter defense problem in linear environments with a single defender, focusing on deterministic and randomized algorithms, respectively. Chapter 4 considers a perimeter defense problem in conical environments with multiple mobile defenders, and Chapter 5 addresses a perimeter defense problem with heterogeneous defenders. Chapter 6 considers a novel motion planning problem of a Dubins-Laser System and Chapter 7 considers a surveillance problem with two trackers and a single defender. Finally, Chapter 8 summarizes this work and discusses some possible extensions.

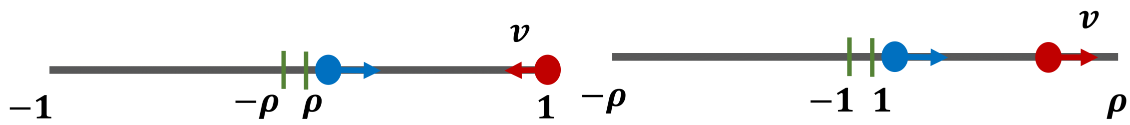
# CHAPTER 2

# COMPETITIVE PERIMETER DEFENSE IN LINEAR ENVIRONMENTS

This chapter addresses a problem in which a robotic defender, having a maximum unit speed, is tasked with defending a given set of points, termed as the perimeter from mobile intruders. The intruders are released in an *arbitrary manner* at specified locations. Upon release, each intruder moves with a fixed speed $v < 1$ towards the perimeter. The defender seeks to intercept maximum number of intruders before the intruders reach the perimeter. We consider two possible scenarios: (i) the intruders are released at the endpoints of the line segment and move inwards, i.e., towards the midpoint of the environment, or (ii) the intruders are released in the interior of the line segment and move outwards, i.e., away from the midpoint of the environment. We focus on design and analysis of *online* algorithms to route the defender as the input, consisting of release times, number of intruders and locations of intruders, is unknown a priori. This work is motivated by the potential use of a single drone to relay real-time information about a road hazard (perimeter) to vehicles driving on both sides of a highway. Military applications include situational awareness using a drone around defense assets in narrow environments, e.g., boats traversing a river, or convoys moving in a valley.

This chapter presents a competitive analysis approach to perimeter defense accounting for worst case inputs. We do not impose any assumption on the number or the release process of the intruders. The contributions of this chapter are three-fold:

1. **Perimeter defense model with two scenarios:** This chapter considers a perimeter defense problem in two scenarios: 1) Inward moving intruders and 2) Outward moving intruders. In particular, both scenarios can be specified using two parameters: (i) the speed ratio $v \in (0, 1]$ between the intruders and the defender and, (ii) the ratio $\rho > 0$, which denotes the size of the perimeter (distance between the points of interest) relative to the size of the environment.




(a) *Incoming intruderss* (**P1**): The intruders move inwards with speed $v$.

(b) *Escaping intruders* (**P2**): The intruders move outwards with speed $v$.

Figure 2.1 Description of the environment $\mathcal{E}(\rho)$. The vehicle is shown as a blue dot and the red dots depict the intruders.

2. **Necessary conditions:** For each scenario, we prove necessary conditions on the competitiveness of any algorithm. Specifically, we characterize a parameter regime of $v$ and $\rho$ in which no algorithm can be $c$-competitive for any constant $c \geq 1$, and a second parameter regime in which every algorithm is at best 2-competitive.

3. **Algorithm design and analysis:** For incoming moving intruders, we design and analyze three classes of algorithms and establish their competitive ratios. Specifically, the first two algorithms are provably 1 and 2-competitive, respectively. The third is a family of algorithms, each of which covers a specific parameter regime with a specified competitive ratio in that parameter regime. Finally, for outward moving intruders, we propose and analyze two algorithms and characterize parameter regimes for which they are provably 1 and 4-competitive, respectively.

This chapter, content of which is based on [9, 8], is organized as follows. In Section 2.1, we formally define our problem. Section 2.2 presents the algorithms and analysis for inward moving intruders. Section 2.3 extends our algorithms and analysis to outward moving intruders. Section 2.4 presents and discusses the parameter regime plots for the two scenarios. Finally, section 2.5 summarizes this chapter.

## 2.1 Problem Formulation

In this work, we consider a perimeter defense problem in a linear environment $\mathcal{E}(\rho)$ in which intruders arrive over time at either position $-1$ or 1 and move with fixed speed $v$ towards the perimeter, defined by the two points $-\rho$ and $\rho$, respectively. To avoid a trivial problem formulation, we will suppose that $\rho \neq 1$. There are two possibilities for $\mathcal{E}(\rho)$. If

$\rho < 1$, then $\mathcal{E}(\rho) = [-1, 1]$, and we define the *incoming intruders* problem variant **P1** (Figure 2.1a) where the intruders move inwards to reach the perimeter. If $\rho > 1$, then $\mathcal{E}(\rho) = [-\rho, \rho]$, and we define the *escaping intruders* problem variant **P2** (Figure 2.1b) where the intruders move outwards to reach the perimeter. In both variants, the intruders are located between $\rho$ and 1 or $-\rho$ and $-1$. In both variants, a single defender with simple motion and a maximum speed of unity is used to defend the perimeter. The defender *captures* an intruder when its location is the same as that of the intruder. We assume that the defender captures an intruder instantaneously, so it can capture an intruder while moving. An intruder is said to be *lost* by the defender if the intruder reaches the perimeter without being captured. Finally, we assume that the defender is located at the origin at time 0, unless otherwise specified.

Although the online and the optimal offline algorithms compute the velocity of the defender, it turns out that specifying the direction in which the defender must move is more important than the speed of the defender. In fact, we will formally establish in Lemma 4 that it is not advantageous for the defender to move with speed less than 1 in any interval of time; that is, the defender either does not move or it moves with speed 1.

**Problem Statement:** Our aim is to design fundamental limits on achieving finite $c$-competitiveness and deterministic $c$-competitive algorithms for problems **P1** and **P2**.

Let *extreme speed algorithms* $\mathcal{A}'$ be the algorithms that always move the defender with unit speed or keep it stationary. Further, let *non-adaptive* input instances be the inputs in which the arrival of intruders is not based on the movement of the defender. Then, due to the following result, we restrict our attention to *extreme speed algorithms* in this work.

**Lemma 4** (Extreme speed algorithms)**.** *Given any problem instance* $\mathcal{P}(\rho, v)$*, for any non-adaptive input instances, extreme speed algorithms are no worse than general algorithms which can move with any speed in the range* $[-1, 1]$.

*Proof.* Let $\mathcal{A}$ denote any general speed algorithm and $I$ denote an arbitrary non-adaptive input instance in which there are $n$ distinct time points where $\mathcal{A}$ captures at least one intruder of $I$. We define the *capture profile* of $\mathcal{A}$'s execution on $I$ as the set of pairs $(x_i, k_i)$

for $1 \leq i \leq n$ where $k_i$ is the time of the $i$th capture and $x_i$ is the location where the $i$th capture occurred. Let $I(x_i, k_i)$ denote the intruders in $I$ captured by $\mathcal{A}$ at location $x_i$ at time $k_i$. We add to this capture profile the pair $(x_0, k_0)$ which denotes the starting location for the defender and the starting time instant. Our goal is to show that there exists an extreme speed algorithm $\mathcal{A}'$ that can match $\mathcal{A}$'s capture profile for $I$: namely having the defender at location $x_i$ at time $k_i$ for $0 \leq i \leq n$. If we can show this, then since $I$ is non-adaptive, $\mathcal{A}'$ will capture the intruders in $I(x_i, k_i)$ for $1 \leq i \leq n$ at $x_i$ at time $k_i$ and the result follows.

We now show that there exists an $\mathcal{A}'$ that can match $\mathcal{A}$'s capture profile for $I$. We prove this by induction on $i$; namely, that there exists an $\mathcal{A}'$ that will have the defender at point $x_j$ at exactly time $k_j$ for $0 \leq j \leq i$ for $0 \leq i \leq n$. The base case where $i = 0$ is straightforward as the defender starts at location $x_0$ at time $k_0$ for both algorithms. We now show that if this holds up to $i$, then it also holds for $i + 1$. Given the induction hypothesis, we know that there exists an $\mathcal{A}'$ that will have the defender at point $x_j$ at exactly time $k_j$ for $0 \leq j \leq i$. We extend that by showing that $\mathcal{A}'$ can also have the defender at $x_{i+1}$ at time $k_{i+1}$.

Let $V_{\mathcal{A}} \in [-1, 1]$ denote the speed with which $\mathcal{A}$ moves its defender. Then, at time $k_i$, $\mathcal{A}'$ keeps the defender stationary at $x_i$ for $\frac{(x_{i+1}-x_i)(1-V_{\mathcal{A}})}{V_{\mathcal{A}}}$ amount of time. $\mathcal{A}'$ then moves the defender with unit speed to location $x_{i+1}$ arriving exactly at time $k_{i+1}$. This completes the inductive step and, in turn, completes the proof. □

We now provide an example for problem **P1** with $\rho = v = 0.5$ to elucidate the use of the competitive analysis technique for this work. An analogous example can be constructed for problem **P2**.

### 2.1.1 Simple Example

For this example, we use a natural algorithm, First-Come-First-Served (FCFS), defined as follows:

**FCFS Algorithm:** FCFS moves the defender with unit speed towards the earliest intruder that is released in the environment which is not lost or captured by the defender. If intruders arrive at the same time instant, then FCFS breaks ties arbitrarily.

Let's first consider an input instance $I_1$ consisting of a single intruder released at endpoint $+1$ a time 0. FCFS can capture the single intruder of $I_1$ at location $2/3 \approx 0.66$. Similarly, an optimal offline algorithm $\mathcal{O}$ also captures the single intruder. Although $n_{\mathcal{O}}(I_1) = n_{\mathcal{FCFS}}(I_1)$, this does not mean that FCFS has a competitive ratio of 1. Such a result requires FCFS to be 1-competitive for all input instances.

The adversary, which knows the online algorithm a priori and can design the input instance based on this knowledge, would choose a more aggressive input instance $I_2$, which has an intruder arriving at endpoint $+1$ at time instant 0 and a burst of $x + 1$ intruders arriving at endpoint $-1$ at time $\epsilon$, where $\epsilon > 0$ is a very small number. Against $I_2$, FCFS captures only the first intruder which arrives at endpoint $+1$. This is because the total time required by the defender following FCFS to reach $-\rho$ after capturing the first intruder is $2/3 * 2 + 0.5 \approx 1.833$. On the other hand, the $x + 1$ burst intruders reach $-\rho$ at time $1 + \epsilon$. Meanwhile, optimal offline algorithm $\mathcal{O}$, knowing $I_2$ at time 0, will move its defender towards endpoint $-1$ at time 0 and capture the $x + 1$ burst intruders sacrificing the first intruder that is released at endpoint $+1$. Thus, $n_{\mathcal{O}}(I_2) = (x + 1)n_{\mathcal{FCFS}}(I_2)$. The existence of input instance $I_2$ proves that FCFS is not $c$-competitive for any $c > 1$ when $\rho = v = 0.5$ as the adversary can set $x = c$. This concludes the example and leads to the following remark, which is established formally in [9].

**Remark 1.** *For any problem instance $\mathcal{P}$ for which $\frac{2}{v+1} + \rho > \frac{1-\rho}{v} + \epsilon$ for some small $\epsilon > 0$, FCFS is not $c$-competitive for any constant $c$.*

We now briefly explain our motivation for designing multiple algorithms. The effectiveness of each of our algorithm is characterized by (i) its competitive ratio $c$ and (ii) the parameter regime, i.e., the set of values of parameters $\rho$ and $v$, in which the algorithm is $c$-competitive. For each algorithm that we present, there exists a parameter regime where it provides the smallest competitive ratio. For example, in Section 2.2, we present algorithms Sweep that is 1-competitive and CaC that is 2-competitive. Clearly, for parameter regimes where both are effective, Sweep should be used since it has a lower competitive ratio. However, there exist parameter regimes where CaC is 2-competitive and Sweep is not $c$-competitive, so CaC should be used in such situations. Note that in Section 2.2, the parameter regimes are described mathematically as they correspond to equations of the problem parameters. To better visualize the regimes, we provide numerical plots in Section 2.4. Similar explanations hold for algorithms designed in Section 2.3.

## 2.2 Incoming Intruders

This section addresses problem **P1** in which $\rho < 1$, i.e., the physical region $\mathcal{R}(\rho) = [-1, 1]$ and the intruders move inwards with fixed speed $0 < v < 1$. We first establish two necessary conditions on achieving a finite competitive ratio and then design three classes of online algorithms with provably finite competitive ratios.

### 2.2.1 Fundamental Limits

We start with a necessary condition for achieving a $c$-competitive ratio for any constant $c$ followed by a necessary condition for achieving a strictly better than 2-competitive ratio.

**Theorem 5** (Necessary condition for $c$-competitiveness). *For any problem instance $\mathcal{P}(\rho, v)$ such that $v > \frac{1-\rho}{2\rho}$, there does not exist a $c$-competitive algorithm for any finite $c \geq 1$.*

*Proof.* Recall from Definition 1 that an online algorithm $\mathcal{A}$ is $c$-competitive if $\inf_{\gamma \geq 1}\{\gamma : n_{\mathcal{O}}(\mathcal{I}, \mathcal{P}) \leq \gamma\, n_{\mathcal{A}}(\mathcal{I}, \mathcal{P}), \forall \mathcal{I}\}\} = c$. Thus, the idea behind the proof is to construct an input sequence $\mathcal{I}$ such that for any constant $\gamma \geq 1$, the condition $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P}) \leq \gamma n_{\mathcal{A}}(\mathcal{I}, \mathcal{P})$ does not hold for any online algorithm $\mathcal{A}$.

We start by constructing an input sequence for any online algorithm $\mathcal{A}$ and then compare the total number of intruders captured to that of the optimal offline algorithm $\mathcal{O}$. We assume that the defender starts at the origin for both $\mathcal{A}$ and $\mathcal{O}$.

The input sequence has two phases: a stream of intruders that arrive at endpoint 1, 2 time units apart starting at time 1 and a burst of $c+1$ intruders that arrive at endpoint $-1$ at time $t$. Time instant $t$ corresponds to the first time the defender moves to $\rho$ according to $\mathcal{A}$. If the defender never moves to location $\rho$, the stream never ends and the burst never arrives, so $\mathcal{A}$ will not be $c$-competitive for any constant $c$, and the first result follows. Let $i$ be the number of stream intruders released up to and including time $t$; note that $i$ might be 0 if the defender reaches $\rho$ before time 1 when the first stream intruder is released. The defender, following $\mathcal{A}$, will capture at most intruder $i$. In particular, because the stream intruders arrive 2 time units apart, all $i-1$ stream intruders have reached $\rho$ before time $t$. Furthermore, since $v > \frac{1-\rho}{2\rho}$, the defender will not be able to capture the burst of $c+1$ intruders that arrive at time $t$. We now determine the number of intruders $\mathcal{O}$ can capture. Two cases arise: $i = 0$ and $i > 0$.

*Case 1:* If $i = 0$, then this means that the defender reached location $\rho$ before time 1, so $t < 1$. $\mathcal{O}$ can move the defender to location $-\rho$ by time $t$ and thus capture all $c+1$ burst intruders, and the result follows. *Case 2:* If $i > 0$, then this means that the defender, following $\mathcal{A}$, reached location $\rho$ no earlier than time 1, so $1 \leq t$. In this case, $\mathcal{O}$ can capture the first $i-1$ stream intruders immediately upon arrival by moving the defender to location 1 at time 1 and staying there until the first $i-1$ stream intruders have been captured. $\mathcal{O}$ can then move the defender to position $-\rho$ before the burst intruders have been released since the stream intruders arrive 2 time units apart. Note that if $i = 1$, $\mathcal{O}$ moves the defender immediately to $-\rho$. Therefore, from Definition 1, we obtain the result.

In either case, we show that $\mathcal{O}$ can capture all the burst intruders and at least all but the last stream intruder whereas $\mathcal{A}$ will capture at most 1 stream intruder, and the result follows. □

**Corollary 1.** *For any problem instance $\mathcal{P}(\rho, v)$ such that $v > \frac{1-\rho}{2\rho}$, no algorithm (online or offline) can capture all intruders.*

*Proof.* The result follows by observing that in the input sequence $\mathcal{I}$, constructed for the proof of Theorem 5, there are choices for $t$ including $t = 1$ such that no algorithm can capture all $i$ stream intruders and all $c+1$ burst intruders, where $t$ and $i$ are analogously defined as in proof of Theorem 5. This concludes the proof. □

While the previous result provides a fundamental limit on the existence of a $c$-competitive algorithm, the following result provides a necessary condition to achieve a strictly better than 2-competitive ratio for an ecosystem.

**Theorem 6** (Necessary condition for $c(\mathcal{P}) < 2$)**.** *Given any problem instance $\mathcal{P}$, $c(\mathcal{P}) < 2$ only if $v < \frac{1-\rho}{1+\rho}$.*

*Proof.* In this proof, all of our input sequences consist of two intruders, $a$ and $b$, where $a$ arrives at location -1 and $b$ arrives at location 1. We assume that the defender starts at the origin at time 0. We first consider the case where $v = \frac{1-\rho}{1+\rho}$. Consider the following input instance $\mathcal{I}_1$ where both $a$ and $b$ arrive at time 1. For input sequence $\mathcal{I}_1$, we describe one of the two symmetric solutions below. At time 0, the defender moves toward location 1 capturing intruder $b$ at location 1 at exactly time 1. The defender then moves to location $-\rho$ to capture intruder $a$ at location $-\rho$ at exactly time $2+\rho$. The defender has just enough time to do this given the condition that $v = \frac{1-\rho}{1+\rho}$ which is equivalent to $1+\rho = \frac{1-\rho}{v}$. Thus, any algorithm that hopes to be better than 2-competitive must capture both intruders in $\mathcal{I}_1$, and the only way to do so is to move immediately to location -1 or 1 reaching at exactly time 1.

Now consider input sequences $\mathcal{I}_2$ and $\mathcal{I}_3$ which also consists of two intruders. In $\mathcal{I}_2$, intruder $a$ arrives at time 1 and intruder $b$ arrives at time $1+\epsilon$ where $\epsilon < 2\rho v$. In $\mathcal{I}_3$, intruder $b$ arrives at time 1 and intruder $a$ arrives at time $1+\epsilon$ where $\epsilon < 2\rho v$. Online algorithms which have the defender arriving at location 1 at time 1 can capture at most one of the two intruders in input instance $\mathcal{I}_2$. This follows as the defender can only capture intruder $a$ if it moves immediately to arrive at location $-\rho$ at time $2+\rho$. However, because $\epsilon < 2\rho v$ and $v = \frac{1-\rho}{1+\rho}$, the defender cannot intercept intruder $b$ before $b$ passes location $\rho$. Similarly, algorithms which have the defender arrive at location -1 at time 1 can capture at most one of the two intruders in input instance $\mathcal{I}_3$. At the same time, there exists an optimal offline algorithm having future information of the input sequences which captures both intruders in input instance $\mathcal{I}_2$ and a different algorithm which captures both intruders in $\mathcal{I}_3$ by simply moving to the correct location at time 1 and then intercept the other intruder before it reaches $+\rho$ or $-\rho$.

We now consider the case where $v > \frac{1-\rho}{1+\rho}$. Now we only need two input sequences $\mathcal{I}_4$ and $\mathcal{I}_5$. In $\mathcal{I}_4$, intruder $a$ arrives at time 1 and intruder $b$ arrives at time $1+\epsilon$ where $\epsilon = 1+\rho-\frac{1-\rho}{v}$. In $\mathcal{I}_5$, intruder $b$ arrives at time 1 and intruder $a$ arrives at time $1+\epsilon$ where $\epsilon = 1+\rho-\frac{1-\rho}{v}$. Because $v > \frac{1-\rho}{1+\rho}$, $\epsilon > 0$. Using similar arguments to those for $\mathcal{I}_2$ and $\mathcal{I}_3$, it follows that no single online algorithm can capture both intruders in both $\mathcal{I}_4$ and $\mathcal{I}_5$.

In summary, even restricting the set of possible input sequences to $\{\mathcal{I}_1, \ldots, \mathcal{I}_5\}$, no single online algorithm can capture both intruders from all five input sequences, but since there exist an optimal offline algorithms which capture both intruders for all five input sequences, it follows that $c(\mathcal{P}) \geq 2$ when $v \geq \frac{1-\rho}{1+\rho}$ and the result is established. □

First-Come-First-Served (FCFS) is defined as the algorithm which moves the defender at unit speed towards the earliest intruder to arrive that is not lost (guaranteed to reach its goal before the defender) or already captured breaking ties arbitrarily. We now show that FCFS is not a $c$-competitive algorithm.

**Lemma 7.** *For any $\mathcal{P}$ where $\frac{2}{v+1} + \rho > \frac{1-\rho}{v} + \epsilon$ for some small $\epsilon > 0$, FCFS is not $c$-competitive for any constant $c$.*

*Proof.* We prove this result by constructing an adversarial input sequence against FCFS. Consider the following input sequence $\mathcal{I}(c)$. Let the first intruder be released at time 0 at point 1. Let $c+1$ intruders be released at time $\epsilon$ at point -1. FCFS will move the defender immediately towards the first intruder and intercept it at point $\frac{1}{v+1}$ at time $\frac{1}{v+1}$. Because of the condition $\frac{2}{v+1} + \rho > \frac{1-\rho}{v} + \epsilon$, FCFS cannot get the defender to position $-\rho$ before the $c+1$ intruders released at time $\epsilon$ reach position $-\rho$. It follows that $n_{FCFS}(\mathcal{I}(c), \mathcal{P}) = 1$. On the other hand, if the defender had moved toward position $-\rho$ immediately, it would capture those $c+1$ intruders. Thus, $n_{\mathcal{O}}(\mathcal{I}(c), \mathcal{P}) = c+1$, and the result follows.

This result generalizes to any variation of FCFS which services the first arriving intruder, if possible, before any later arriving intruder. □

With these necessary conditions in place, we now turn our focus to the design of algorithms with provable guarantees on the competitive ratio.

### 2.2.2 Algorithms

We now describe and analyze three algorithms, characterizing parameter regimes with provably finite competitive ratios for incoming intruders.

#### 2.2.2.1 Sweeping Algorithm

We define the Sweeping algorithm (Sweep) as follows. At initial time, Sweep moves the defender with maximum speed towards location 1. Hereafter, the defender changes direction only when it reaches an endpoint 1 or $-1$. Once the defender reaches an endpoint, it moves with maximum speed towards the opposite endpoint. Sweep is an *open-loop* algorithm; meaning, Sweep does not require any information about intruders. One logical modification is to stop moving in a given direction if there are no intruders in that direction. We establish that, even with this modification, Sweep achieves the same competitive ratio.

**Theorem 8.** *For any problem instance $\mathcal{P}$, Sweep is* 1*-competitive if $v \leq \frac{1-\rho}{3+\rho}$. If not, Sweep is not $c$-competitive for any constant $c$.*

*Proof.* Consider that $v \leq \frac{1-\rho}{3+\rho}$ holds. Any intruder will take $\frac{1-\rho}{v}$ time to get from its arrival location to the perimeter. Without loss of generality, consider an intruder $i$ that arrives at location 1 and thus, takes $\frac{1-\rho}{v}$ time to reach $\rho$. In the worst case, the defender will have left 1 just before intruder $i$ arrived. In the worst case, the defender will take $3+\rho$ time to reach $\rho$ moving towards intruder $i$. Since $3+\rho \leq \frac{1-\rho}{v}$, Sweep's defender will reach $\rho$ before the intruder $i$ and thus, intruder $i$ will be captured. This concludes the proof for the first result.

For $v > \frac{1-\rho}{3+\rho}$ there exists an input sequence where intruders only arrive at endpoint 1 just after the defender has left 1. As $v > \frac{1-\rho}{3+\rho}$, all intruders will be lost and the second result follows. □

Note that the first result holds for the modified Sweep where it stops moving in a given direction if there are no intruders in that direction. For the second result, we introduce some intruders at $-1$ to ensure that the modified Sweep will move the defender towards $-1$. The result still holds, even though the intruders released at location $-1$ might be captured, by releasing more intruders at location 1.

The following definition will be helproofful for the analysis of the subsequent algorithms. A set of intruders $S$ is said to be on the *same side* as the defender if the defender is located at $\rho$ (resp. $-\rho$) and $S \subset (\rho, 1]$ (resp. $[-1, -\rho)$). Similarly, we say that $S$ is on the *opposite side* of the defender if the defender is located at $\rho$ (resp. $-\rho$) and $S \subset [-1, -\rho]$ (resp. $[\rho, 1]$). For $i \geq 1$, $S^i_{\text{opp}}$ and $S^i_{\text{same}}$ denotes the set of intruders in an input instance $I$ that arrive in the $i$th time interval that are on the opposite side and same side of the defender, respectively and $|S|$ denotes the cardinality of $S$.

#### 2.2.2.2 Compare and Capture algorithm

We now present a Compare and Capture (CaC) algorithm that is provably 2-competitive beyond the parameter regime of the Sweep algorithm. CaC is not open-loop, but is *memoryless*, i.e., the actions of CaC depend only on the present state of the defender and the intruders.

An epoch $k$ for the CaC algorithm is the time interval when the defender moves from location $x_k$ to location $x_{k+1}$ and is about to move from $x_{k+1}$, capturing some intruders on the way. Location $x_k$ is always either $\rho$ or $-\rho$. We denote the beginning of epoch $k$ by $k_S$. For epoch $k$, we define $S^k_{\text{same}}$ as the set of intruders on the *same side* as the defender at time $k_S$, and $S^k_{\text{opp}}$ as the set of intruders on the *opposite side* of the defender that are between $\rho + 2\rho v$ and $\rho + 2v\rho + \frac{2v(1-\rho)}{1+v}$ away from the origin at time $k_S$. More precisely, if the defender is located at $x_k = \rho$, then $S^k_{\text{opp}}$ is defined as the set of intruders contained in $[-(\rho + 2\rho v + \frac{2v(1-\rho)}{1+v}), -(\rho + 2\rho v)]$.

The CaC algorithm, summarized in Algorithm 1, works as follows: At epoch $k$, for any

**Algorithm 1:** Compare-and-Capture Algorithm

Defender waits for $\frac{1-\rho-3\rho v}{v}$ time units at the center
**if** *intruders in* $[\rho+3\rho v, 1] \leq$ *intruders in* $[-1, -(\rho+3\rho v)]$ **then**
  Move to $-\rho$
**else**
  Move to $\rho$.
**end**
**for** *each epoch* $k \geq 1$ **do**
  **if** *at epoch* $k$, $|S^k_{\text{same}}| < |S^k_{\text{opp}}|$ **then**
    Move to capture all intruders in $S^k_{\text{opp}}$
    Move to $x_{k+1} = -x_k$
  **else**
    Move to capture intruders in $S^k_{\text{same}}$.
    Move to $x_{k+1} = x_k$
  **end**
**end**

$k \geq 1$, CaC computes the total number of intruders located in $S^k_{\text{same}}$ and $S^k_{\text{opp}}$. If the total number of intruders in $S^k_{\text{opp}}$ is more than the total number of intruders in $S^k_{\text{same}}$, then the defender moves, in the direction of the origin, for at most $2\rho + \frac{4v(1-\rho)}{(1+v)^2}$ time to capture all intruders located in $S^k_{\text{opp}}$ and then returns to $x_{k+1} = -x_k$. Otherwise, the defender moves away from the origin for at most $\frac{1-\rho}{1+v}$ time to capture all intruders from the set $S^k_{\text{same}}$ and then returns to $x_{k+1} = x_k$.

For the initial case, CaC waits at the origin until the first intruder that arrives is $3\rho v$ distance away from the perimeter. If the total number of intruders located in $[\rho + 3\rho v, 1]$ is more than the total number of intruders located in $[-1, -(\rho + 3\rho v)]$, then the defender moves to $x_1 = \rho$. If not, the defender moves to $x_1 = -\rho$. The first epoch starts when the defender reaches $x_1$.

To establish 2-competitiveness of CaC, we first establish that any intruder not belonging to $S^k_{\text{same}}$ or $S^k_{\text{opp}}$ in an epoch $k$ will not be lost during epoch $k$. At the start of epoch $k$, we say that an intruder lies outside of the set $S^k_{\text{opp}}$, if the defender is located at $\rho$ (resp. $-\rho$) and the intruder is contained in $[-1, -(\rho + 2\rho v + \frac{2v(1-\rho)}{1+v}))$ (resp. $(\rho + 2\rho v + \frac{2v(1-\rho)}{1+v}, 1])$.

**Lemma 9.** *In every epoch $k$, any intruder that lies outside of the set $S^k_{\text{same}}$ and $S^k_{\text{opp}}$ is not* *lost if* $\frac{\rho v}{1-\rho} + \frac{v^2}{(1+v)^2} \leq \frac{1}{4}$.

*Proof.* Without loss of generality, consider that the defender is located at $x_k = \rho$ at time $k_S$. Two cases arise:

*Case 1:* $|S^k_{\text{same}}| \geq |S^k_{\text{opp}}|$*:* In this case, the defender moves away from the origin to capture intruders located in $S^k_{\text{same}}$. The defender takes at most $\frac{1-\rho}{1+v}$ time to capture these intruders. The total time taken by the defender to capture the intruders and return to $x_{k+1} = x_k$ in epoch $k$ is at most $\frac{2(1-\rho)}{1+v}$. Since the intruders in $S^k_{\text{opp}}$ are located in $[-(\rho+2\rho v+\frac{2v(1-\rho)}{1+v}), -(\rho+2\rho v)]$, any intruder in the opposite side which was outside the set $S^k_{\text{opp}}$ will be at least $2\rho v$ distance away from $\rho$ and thus, will be contained in $S^{k+1}_{\text{opp}}$.

*Case 2:* $|S^k_{\text{same}}| < |S^k_{\text{opp}}|$*:* The total time taken by the defender to capture intruders located in $S^k_{\text{opp}}$ and return back to $x_{k+1} = -x_k$ is at most $2\rho + \frac{4v(1-\rho)}{(1+v)^2}$. In order to ensure that any intruder that did not belong to $S^k_{\text{same}}$ is at least $2\rho v$ distance away from $\rho$ at the end of epoch $k$, we require

$$2\rho + \frac{4v(1-\rho)}{(1+v)^2} \leq \frac{1-\rho-2\rho v}{v} \Rightarrow \frac{\rho v}{1-\rho} + \frac{v^2}{(1+v)^2} \leq \frac{1}{4}.$$

This concludes the proof. □

**Lemma 10.** *In each epoch* $k$ *of CaC,* $S^k_{\text{opp}}$ *is well defined if* $\rho + 2\rho v + \frac{2v(1-\rho)}{1+v} \leq 1$.

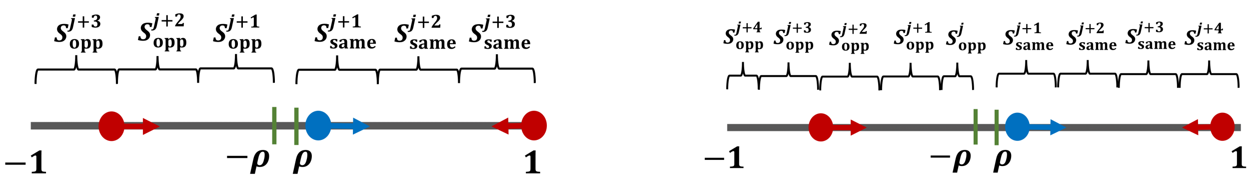
*Proof.* In order to ensure that $S^k_{\text{opp}}$ is well defined, we require that $S^k_{\text{opp}}$ is contained in the environment, i.e., $S^k_{\text{opp}}$ does not extend beyond 1 or $-1$. Mathematically, this corresponds to $\rho + 2\rho v + \frac{2v(1-\rho)}{1+v} \leq 1$, and the result follows. □

**Theorem 11.** *Compare and Capture algorithm defined in Algorithm 1 is* 2*-competitive for any problem instance* $\mathcal{P}$ *in parameter regime for which both Lemma 9 and Lemma 10 hold.*

*Proof.* First, observe that CaC is well-defined as, from Lemma 10, $S^k_{\text{opp}}$ is well-defined in every epoch $k$. Lemma 9 ensures that every intruder will belong to $S^k_{\text{same}}$ or $S^k_{\text{opp}}$ for some epoch $k$. Due to the comparison of the number of intruders in Algorithm 1, in each epoch, the number of intruders lost is at most the number of intruders captured. Recall that an

**Algorithm 2:** Capture with Patience Algorithm

**Input:** Parameter $i$ such that $2i$ is a positive integer
**Assumption:** Defender stays at origin until time $2\rho$, same side is right of origin
At time $2\rho$, **if** $|S_{\text{opp}}^1| + \cdots + |S_{\text{opp}}^{\lceil i \rceil}| > |S_{\text{same}}^1| + \cdots + |S_{\text{same}}^{\lceil i \rceil}|$ **then**
  Move to $x_2 = -\rho$, same side becomes left of origin
**else**
  Move to $x_2 = \rho$, same side unchanged
**end**
Wait until time $z = \frac{1-\rho}{v}$
**for** *each time instant* $z + 2j'\rho + 2\rho j/i, j, j' \geq 0$ **do**
  **if** $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| > |S_{\text{same}}^{j+1}| + \cdots + |S_{\text{same}}^{j+2i+1}|$ **then**
    Move to $x_{j+1} = -x_j$, same side changes
    Stay at $x_{j+1}$ and capture $S_{\text{opp}}^{j+\lfloor i \rfloor+1}$
    $j' = j' + 1$, $j = j + 1$
  **else**
    Stay at $x_j$ and capture interval $S_{\text{same}}^{j+1}$
    $j = j + 1$
  **end**
**end**




(a) Intervals of length $2\rho v$ ($i = 1$)  (b) Intervals of length $2\rho v/1.5$ ($i = 1.5$)

Figure 2.2 Breakdown of the physical region into regions of length (a) $2\rho v$ ($i = 1$) and (b) $2\rho v/1.5$ ($i = 1.5$) or equivalently time into intervals of length $2\rho$ ($i = 1$) and $2\rho/1.5$ ($i = 1.5$) by the CAP algorithm.

intruder is not considered lost in epoch $k$ if it is contained in $S_{\text{same}}^{k+1}$ or $S_{\text{opp}}^{k+1}$ in the next epoch $k+1$. Thus, CaC captures at least half of the total number of intruders. This concludes the proof. □

#### 2.2.2.3 Capture with Patience Algorithm

We now describe a family of algorithms that we call Capture with Patience (CAP) which covers a larger parameter regime than the previous algorithms at the expense of a higher competitive ratio. CAP is formally defined in Algorithm 2 and is described as follows.

The basic idea behind CAP is to label intruders by the time interval that they arrive in as depicted in Figure 2.2. CAP defines the initial time to be the time instant when the first intruder arrives. CAP has the defender stay at $\rho$ or $-\rho$ and compares a specific interval on the opposite side of the origin with some number of intervals on the same side of the origin. If the opposite side interval contains more intruders than all of the same side intervals collectively, then CAP moves the defender to the opposite side and captures that interval. Otherwise, CAP keeps the defender where it is and captures the next interval on the same side. After capturing an interval, it makes the same comparison to decide which interval to capture next. If an interval represents $t$ units of time, then its length in Figure 2.2 is $tv$ since that is how far an intruder will move in $t$ time units.

We can make CAP have a lower competitive ratio by choosing bigger intervals. CAP specifies the interval size using a parameter $i \geq 1/2$ such that $2i$ is an integer. Specifically, the time intervals span for $2\rho/i$ time units and have length $2\rho v/i$ in Figure 2.2. Clearly, a smaller $i$ leads to a larger interval. Furthermore, $i$ denotes the number of intervals the defender will lose if it moves from $\rho$ to $-\rho$ or vice versa. If $2i$ is an even integer ($i$ is an integer), then the intervals are symmetric on both sides of the origin as depicted in Figure 2.2a. If $2i$ is an odd integer, then the intervals are offset by half an interval on one side of the origin as depicted in Figure 2.2b.

A key feature of the problem is the quantity $z = \frac{1-\rho}{v}$ which represents the time required for any intruder originating at 1 or $-1$ to reach the corresponding location $\rho$ or $-\rho$. In order to guarantee a competitive ratio, we require that $z \geq \frac{2\rho(2i+1)}{i}$ which means that at least $2i+1$ intervals exist on each side of the origin. For the smallest $i = 1/2$, $z \geq 8\rho$. This condition is easier to satisfy as $i$ increases asymptotically approaching the condition $z \geq 4\rho$.

After the initial time $z$, CAP operates as follows: At any time instant $2j'\rho + 2j\rho/i + z$ for $j, j' \geq 0$, the defender is stationed at either $-\rho$ or $\rho$. Here $j'$ is used to characterize the amount of time the defender has spent moving from $\rho$ to $-\rho$ or vice versa and $j$ is used to characterize the amount of time the defender has spent capturing intervals. Without loss of generality, consider that the defender is located at $\rho$.

First, note that for any $j \geq 0$, the intruders in $S_{\text{same}}^{j+1}$ are located between $\rho$ and $\rho + 2\rho v/i$, the intruders in $S_{\text{same}}^{j+2}$ are located between $\rho + 2\rho v/i$ and $\rho + 4\rho v/i$, until the intruders in $S_{\text{same}}^{j+2i+1}$ are located between $\rho + 4i\rho v/i$ and $\rho + 2(2i+1)\rho v/i$. Further, because $z \geq \frac{2\rho(2i+1)}{i}$, all the intruders in $S_{\text{same}}^{j+2i+1}$ have arrived by time $2j'\rho + 2j\rho/i + z$. Analogous conclusions can be drawn for the intruders in $S_{\text{opp}}^{j+1}$, $S_{\text{opp}}^{j+2}$, $\dots$, $S_{\text{opp}}^{j+2i+1}$ when $2i$ is even and $S_{\text{opp}}^{j}$, $S_{\text{opp}}^{j+1}$, $\dots$, $S_{\text{opp}}^{j+2i+1}$ when $2i$ is odd. Denote $\lceil x \rceil$ and $\lfloor x \rfloor$ as the ceil and the floor functions on a real number $x$, respectively. If $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| > |S_{\text{same}}^{j+1}| + |S_{\text{same}}^{j+2}| + \cdots + |S_{\text{same}}^{j+2i+1}|$, then the defender moves to $-\rho$ arriving at time $2(j'+1)\rho + 2j\rho/i + z$, which is just in time to capture all the intruders in $S_{\text{opp}}^{j+\lfloor i \rfloor+1}$ and stays at $-\rho$ to capture all of the intruders in $S_{\text{opp}}^{j+\lfloor i \rfloor+1}$. The defender then reevaluates at time $2(j'+1)\rho + 2(j+1)\rho/i + z$. Otherwise (if $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| \leq |S_{\text{same}}^{j+1}| + \cdots + |S_{\text{same}}^{j+2i+1}|$), the defender stays at $\rho$ and captures all the intruders in $S_{\text{same}}^{j+1}$ and reevaluates at time $2j'\rho + 2(j+1)\rho/i + z$. The key idea is that the defender moves from $\rho$ to $-\rho$ only when it sees sufficient benefit in terms of the number of intruders in $S_{\text{opp}}^{j+\lfloor i \rfloor+1}$ to sacrifice all the intruders in $S_{\text{same}}^{j+1}$, $S_{\text{same}}^{j+2}$, $\dots$, and $S_{\text{same}}^{j+2i+1}$.

For the initial case, the defender stays at the origin until time $2\rho$. At time $2\rho$, if $|S_{\text{opp}}^{1}| + \cdots + |S_{\text{opp}}^{\lceil i \rceil}| > |S_{\text{same}}^{1}| + \cdots + |S_{\text{same}}^{\lceil i \rceil}|$ (intruders to the left of the origin are considered on the opposite side while intruders to the right of the origin are considered on the same side for this special case), then the defender moves to $-\rho$. Otherwise, the defender moves to $\rho$. In either case, the defender then stays at either $-\rho$ or $\rho$ until time $z$.

**Lemma 12.** *Algorithm CAP never moves the defender from $\rho$ to $-\rho$ and then back to $\rho$ (or vice versa) without capturing at least one interval of intruders.*

*Proof.* For the defender to move from $\rho$ to $-\rho$ at time $2j'\rho + 2j\rho/i + z$, the condition $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| > |S_{\text{same}}^{j+1}| + |S_{\text{same}}^{j+2}| + \cdots + |S_{\text{same}}^{j+2i+1}|$ must hold. This implies that $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| > |S_{\text{same}}^{j+2i+1}|$. In order to move directly back to $\rho$ without capturing the intruders in the interval $S_{\text{opp}}^{j+\lfloor i \rfloor+1}$, it must be true that $|S_{\text{same}}^{j+2i+1}| > |S_{\text{opp}}^{j+\lfloor i \rfloor+1}| + |S_{\text{opp}}^{j+\lfloor i \rfloor+2}| + \cdots + |S_{\text{opp}}^{j+\lfloor 3i \rfloor+1}|$. This cannot be true because $|S_{\text{opp}}^{j+\lfloor i \rfloor+1}| > |S_{\text{same}}^{j+2i+1}|$ and the claim is established. ∎

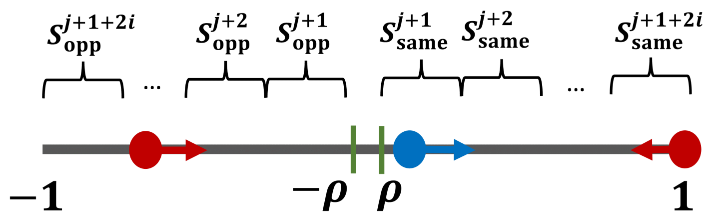



Figure 2.3 Labeling of intervals for an integer $i$.

**Corollary 2.** *For any $j, j' \geq 1$, Algorithm CAP will capture all intruders from one of $S_{\text{same}}^{j}$ or $S_{\text{opp}}^{j+\lfloor i \rfloor}$.*

*Proof.* At time $z + 2(j'-1)\rho + 2\rho(j-1)/i$, the defender is in position to capture $S_{\text{same}}^{j}$ by definition of CAP. If it captures $S_{\text{same}}^{j}$, the claim follows. If not, it moves to the opposite capture point and by Lemma 12 will capture $S_{\text{opp}}^{j+\lfloor i \rfloor}$. □

**Theorem 13.** *The CAP algorithm, which takes an input $i$ and is defined in Algorithm 2 is $2i+2$-competitive for any problem instance $\mathcal{P}$ with $v \leq \frac{i(1-\rho)}{2\rho(2i+1)}$.*

*Proof.* We prove this claim by using a charging scheme where we "charge" lost intervals of intruders to captured intervals of intruders, or equivalently, captured intervals of intruders "pay" for the lost intervals of intruders. We prove this result in two parts. First, we establish that each captured interval is charged at most $2i+1$ times. Second, we establish that each lost interval is paid for by the captured intervals.

Without loss of generality, consider that the defender captures its first and last interval of intruders at $\rho$ and $\rho$, respectively (our proof easily generalizes to an infinite number of intruders). We summarize an execution of CAP with a *trace* which records how many intervals are captured at each location before switching to the opposite location. Formally, we define a trace as a list $[k_{\rho}^{1}, k_{-\rho}^{2}, \ldots, k_{\rho}^{m}]$ wherein we use superscripts to index the elements in the list. From Corollary 2, every element in the trace must be greater than zero with the exception of $k_{\rho}^{1} \geq 0$. A trace fully defines the movement of CAP's defender; namely, it initially moves from 0 to $\rho$, stays at $\rho$ to capture $k_{\rho}^{l}$ intervals, then moves to $-\rho$, stays at $-\rho$ to capture $k_{-\rho}^{2}$, intervals, and so on, until it stays at $\rho$ to capture the final $k_{\rho}^{m}$ intervals.

We now describe how intervals of captured intruders pay for intervals of lost intruders. Each captured interval can be classified into one of two types: type (a) where the defender did not move to capture it, meaning that the defender also captured the previous interval on the same side (or the interval corresponds to the first interval $S^1_{\text{same}}$) or type (b), where the defender did move to capture it, i.e., the defender spent the last $2\rho$ time units moving from $\rho$ to $-\rho$ or vice versa. We establish that both type (a) and type (b) captured intervals are charged at most $2i+1$ times. We first consider type (a) captured intervals. For any arbitrary $l^{th}$, $1 \leq l \leq m$, element in the trace with $k^l_\rho \geq 1$ (resp. $k^l_{-\rho} \geq 1$), note that the last $k^l_\rho - 1$ (resp. $k^l_{-\rho} - 1$) intervals are classified as type (a) intervals. A captured interval $S^{j+q}_{\text{same}}$, where $1 \leq q \leq 2i$ is charged $q$ times to pay for the lost intervals $S^{j+1+\lfloor i \rfloor}_{\text{opp}}, \ldots, S^{j+q+\lfloor i \rfloor}_{\text{opp}}$ (Fig. 2.3). Further, each captured interval $S^{j+q}_{\text{same}}$, where $2i+1 \leq q \leq k^l_\rho$ (resp. $k^l_{-\rho}$) is charged $2i+1$ times to pay for each of the lost intervals $S^{j+q-\lceil i \rceil}_{\text{opp}}, \ldots, S^{j+q+\lfloor i \rfloor}_{\text{opp}}$. Note that if $k^l_\rho \leq 2i$ (resp. $k^l_{-\rho} \leq 2i$), then each captured interval is charged at most $2i$ times.

A type (b) captured interval $S^{j+1+\lfloor i \rfloor}_{\text{opp}}$ (Fig. 2.3) will be charged at most $2i+1$ times – a unit charge to pay all at once for lost intervals $S^{j+1}_{\text{same}}, \ldots, S^{j+1+2i}_{\text{same}}$, and up to $2i$ times to help pay for each of the lost intervals $S^{j+1-\lceil i \rceil}_{\text{opp}}, \ldots, S^{j+\lfloor i \rfloor}_{\text{opp}}$.

To establish the second part of the proof, we decompose a CAP trace into a sequence of *atomic* traces which we define as follows. As we noted earlier, a trace defines the movement of CAP's defender. An atomic trace is an interval characterizing the movement of the defender between $\rho$ and $-\rho$ as well as capture of intruder intervals at $\rho$ and $-\rho$. Specifically, for any $l^{th}$, $2 \leq l \leq m-1$, element of CAP's trace, we define atomic trace $A^l_\rho$ (resp. $A^l_{-\rho}$) as an interval that has the following components:

i An initial component where the defender moves from $-\rho$ (resp. $\rho$) to $\rho$ (resp. $-\rho$).

ii A middle component where the defender captures $k^l_\rho$ (resp. $k^l_{-\rho}$) intervals at $\rho$ (resp. $-\rho$).

iii A final component where the defender returns to $-\rho$ (resp. $\rho$) and completes capturing the first type (b) interval, denoted as $C_b$, in $k_{-\rho}^{l+1}$ (resp. $k_{\rho}^{l+1}$).

Atomic trace $A_{\rho}^{1}$ differs in the initial component as the defender moves from 0 to $\rho$ and the middle component as it may possibly not capture any intervals at $\rho$ if CAP moves the defender to $-\rho$ without capturing an interval at $\rho$. Atomic trace $A_{\rho}^{m}$ has no final component because the defender does not return to $-\rho$. Note that component (iii) for an atomic trace $A_{\rho}^{l}$ (resp. $A_{-\rho}^{l}$) is included in the next atomic trace $A_{\rho}^{l+1}$ (resp. $A_{-\rho}^{l+1}$) (components (i) and (ii)). We include component (iii) in the next atomic trace because we separate how this captured type (b) interval, denoted by $C_b$, is charged to pay for lost intervals. Specifically, in component (iii), $C_b$ is used to pay for up to $2i+1$ lost intervals at the same location as $C_b$ whereas in components (i) and (ii), $C_b$ is used to pay for $2i+1$ lost intervals at the other location. We explain this further below.

We now prove the second part of the proof that the lost intruder intervals are fully paid for by the captured intruder intervals. Specifically, we prove that all the intervals lost at $-\rho$ between those captured in atomic trace $A_{-\rho}^{l-1}$ and atomic trace $A_{-\rho}^{l+1}$ are paid for by the captured intervals in atomic trace $A_{\rho}^{l}$. Likewise, we prove that all the intervals lost at $\rho$ between those captured in atomic trace $A_{\rho}^{l-1}$ and atomic trace $A_{\rho}^{l+1}$ are paid for by the captured intervals in atomic trace $A_{-\rho}^{l}$. We separately handle the boundary cases for the intervals lost at $-\rho$ in atomic traces $A_{\rho}^{1}$ and $A_{\rho}^{m}$. Since the arguments for atomic traces $A_{\rho}^{l}$ and $A_{-\rho}^{l}$ are analogous due to symmetry, we only give the proof for atomic trace $A_{\rho}^{l}$.

We first focus on atomic trace $A_{\rho}^{l}$ where $1 < l < m$. During atomic trace $A_{\rho}^{l}$, the defender loses $k_{\rho}^{l} + 2i$ intervals at $-\rho$. At the start of $A_{\rho}^{l}$, the defender moves from $-\rho$ to $\rho$ which implies that the first interval that the defender captures at $\rho$ in $A_{\rho}^{l}$ pays for the first $2i+1$ lost intervals at $-\rho$. If $k_{\rho}^{l} = 1$, then the captured interval at $\rho$ accounts for all $2i+1 = 2i+k_{\rho}^{l}$ lost intervals at $-\rho$. Thus, for the rest of this proof, we assume that $k_{\rho}^{l} \geq 2$ and show that the remaining $k_{\rho}^{l} - 1$ lost intervals at $-\rho$ are accounted for by the remaining captured intervals in the atomic trace $A_{\rho}^{l}$.

The defender spends $(k_{\rho}^{l} - 1)2\rho/i$ amount of time staying at $\rho$ capturing $k_{\rho}^{l} - 1$ type (a) intervals. Let $t = z + \frac{2\rho(j+k^l_\rho-1)}{i} + 2\rho j'$, where $j, j' > 0$, be the time when the defender finishes capturing the last interval at $\rho$ within $A^l_\rho$. First, because the defender stays at $\rho$ from time $t - (k^l_\rho - 1)2\rho/i$ to time $t - \frac{2\rho}{i}$, the following conditions hold:

$$
\begin{aligned}
|S_{\text{opp}}^{j+k^l_\rho-1+\lfloor i \rfloor}| &\le |S_{\text{same}}^{j+k^l_\rho-1}| + |S_{\text{same}}^{j+k^l_\rho}| + \cdots + |S_{\text{same}}^{j+k^l_\rho-1+2i}|, \text{at time } t - \tfrac{2\rho}{i} \\
|S_{\text{opp}}^{j+k^l_\rho-2+\lfloor i \rfloor}| &\le |S_{\text{same}}^{j+k^l_\rho-2}| + |S_{\text{same}}^{j+k^l_\rho-1}| + \cdots + |S_{\text{same}}^{j+k^l_\rho+2i-2}|, \text{at time } t - \tfrac{4\rho}{i} \\
&\vdots \\
|S_{\text{opp}}^{j+1+\lfloor i \rfloor}| &\le |S_{\text{same}}^{j+1}| + |S_{\text{same}}^{j+2}| + \cdots + |S_{\text{same}}^{j+2i+1}|, \text{at time } t - \tfrac{(k^l_\rho-1)2\rho}{i}.
\end{aligned}
$$

For ease of presentation, we do not update $j$ after each $2\rho/i$ time interval in the above recursive conditions. As the defender moves from location $-\rho$ to $\rho$ and captures the first interval out of $k^l_\rho$ before making its first comparison, $S^{j+1}_{\text{same}}$ denotes the second interval out of $k^l_\rho$ that the defender captured at $\rho$. Next, because the defender goes from $\rho$ to $-\rho$ at time $t$, the following condition holds:

$$
|S_{\text{opp}}^{j+k^l_\rho+\lfloor i \rfloor}| > |S_{\text{same}}^{j+k^l_\rho}| + |S_{\text{same}}^{j+k^l_\rho+1}| + \cdots + |S_{\text{same}}^{j+k^l_\rho+2i}|. \tag{2.1}
$$

We first consider that $k^l_\rho - 1 \ge 2i + 1$. Since $k^l_\rho - 1 \ge 2i + 1$, the first lost interval $S_{\text{opp}}^{j+1+\lfloor i \rfloor}$ is accounted for by the $2i + 1$ captured intervals at $\rho$ as the condition $|S_{\text{opp}}^{j+1+\lfloor i \rfloor}| \le |S^{j+1}_{\text{same}}| + |S^{j+2}_{\text{same}}| + \cdots + |S^{j+2i+1}_{\text{same}}|$ holds. Then, every subsequent captured interval at $\rho$ will account for one lost interval at $-\rho$ as the following conditions hold:

$$
\begin{aligned}
|S_{\text{opp}}^{j+2+\lfloor i \rfloor}| &\le |S^{j+2}_{\text{same}}| + |S^{j+3}_{\text{same}}| + \cdots + |S^{j+2i+2}_{\text{same}}| \\
|S_{\text{opp}}^{j+3+\lfloor i \rfloor}| &\le |S^{j+3}_{\text{same}}| + |S^{j+4}_{\text{same}}| + \cdots + |S^{j+2i+3}_{\text{same}}| \\
&\vdots \\
|S_{\text{opp}}^{j+k^l_\rho-1-2i+\lfloor i \rfloor}| &\le |S_{\text{same}}^{j+k^l_\rho-1-2i}| + |S_{\text{same}}^{j+k^l_\rho-2i}| + \cdots + |S_{\text{same}}^{j+k^l_\rho-1}|.
\end{aligned}
$$

We now account for the last $2i$ lost intervals, $S_{\text{opp}}^{j+k^l_\rho-2i+\lfloor i \rfloor}, \ldots, S_{\text{opp}}^{j+k^l_\rho-1+\lfloor i \rfloor}$, lost at $-\rho$. Consider any arbitrary lost interval $S_{\text{opp}}^{j+q+\lfloor i \rfloor}$, where $k^l_\rho - 2i \le q \le k^l_\rho - 1$. To account for this lost interval, CAP was supposed to capture the $2i + 1$ intervals $S^{j+q}_{\text{same}}, \ldots, S^{j+q+2i}_{\text{same}}$.

CAP actually captures $S_{\text{same}}^{j+q}, \ldots, S_{\text{same}}^{j+k_\rho^l-1}$ but does not capture $S_{\text{same}}^{j+k_\rho^l}, \ldots, S_{\text{same}}^{j+q+2i}$. CAP makes up for these intervals by capturing the interval $S_{\text{opp}}^{j+k_\rho^l+\lfloor i \rfloor}$ which, due to equation (2.1), contains more intruders than all of the intervals $S_{\text{same}}^{j+k_\rho^l}, \ldots, S_{\text{same}}^{j+q+2i}$ combined. By redefining the bounds on $q$ as $1 \leq q \leq 2i+1$, the proof for the case when $k_\rho^l - 1 < 2i+1$ is analogous to when we consider the remaining $2i$ intervals out of $k_\rho^l - 1$, so we omit this case for brevity.

Earlier, we noted that type (b) captured intervals would be charged up to $2i$ times to pay for lost intervals on the same side as the captured intervals. Here we see that the last type (b) interval in atomic interval $A_\rho^l$ is charged exactly $\max\{2i, k_\rho^l - 1\}$ times for lost intruders at $-\rho$.

Thus, we conclude that in an atomic trace of the type $[k_\rho^l, 1]$, all lost intervals have been fully accounted for by the captured intervals. After the defender finishes capturing the interval $S_{\text{opp}}^{j+k_\rho^l+\lfloor i \rfloor}$, note that the next atomic trace $A_{-\rho}^{l+1}$ includes the captured interval $S_{\text{opp}}^{j+k_\rho^l+\lfloor i \rfloor}$. At the start of the next atomic trace $A_{-\rho}^{l+1}$, a new steady state is achieved as all $S_{\text{same}}^{j+k_\rho^l}, \ldots, S_{\text{same}}^{j+k_\rho^l+2i}$ intervals are accounted for since equation (2.1) holds. Further, following the same argument as for atomic trace $A_\rho^l$, it can be shown that all of the lost intervals at $\rho$ from the start of $A_{-\rho}^{l+1}$ until the start of $A_\rho^{l+2}$ are accounted for by the captured intervals of $A_{-\rho}^{l+1}$.

Now it remains to be shown that the lost intervals are fully accounted for in the first and the last atomic trace. We first consider the last atomic trace, i.e., $A_\rho^m$. As the defender moves from $-\rho$ to $\rho$, the first type (b) interval that the defender captures at $\rho$ pays for $2i+1$ lost intervals at $\rho$. If $k_\rho^m = 1$, then the captured interval accounts for all lost intervals at $-\rho$. Thus we now consider that $k_\rho^m > 1$. As the defender does not switch sides, starting from the time instant when the defender finished capturing the first of $k_\rho^m$ intervals, after every $\frac{2\rho}{i}$ time units, the total number of intruders that are on the same, i.e., right side of the defender will be no less than the total number of the intruders on the opposite, i.e., left side of the defender. Thus, all the intruder intervals that are lost at $-\rho$ are accounted for.

We now consider the first atomic trace $A_\rho^1$. As the defender decides to move to $\rho$ at time $2\rho$, the condition $|S^1_{\text{same}}| + \cdots + |S^{\lceil i \rceil}_{\text{same}}| \geq |S^1_{\text{opp}}| + \cdots + |S^{\lceil i \rceil}_{\text{opp}}|$ must hold. If $k^1_\rho = 0$ then this means that at time $z$, the defender starts to move towards $-\rho$ as the condition $|S^{1+\lceil i \rceil}_{\text{opp}}| > |S^1_{\text{same}}| + \cdots + |S^{2i+1}_{\text{same}}|$ holds. Thus, the $2i+1$ lost intervals in $A^1_\rho$ as well as the first $\lceil i \rceil$ intervals at $-\rho$ are accounted for by capturing $S^{1+\lceil i \rceil}_{\text{opp}}$. The proof for $k^1_\rho > 0$ is analogous to the proof for $A^l_\rho, 1 < l < m-1$ and has been omitted for brevity.

Since we have shown our charging scheme pays for all lost intruder intervals and each captured intruder interval pays for at most $2i+1$ lost intruder intervals, the result follows. □

We now establish two lower bounds on the competitive ratio for the CAP algorithm including showing that the bound is tight for $i = 0.5$ and $i = 1$.

**Lemma 14.** *Algorithm CAP is no better than $2i+1$-competitive for $v \leq \frac{i(1-\rho)}{2\rho(2i+1)}$. For $i = 0.5$ (resp. $i = 1$), Algorithm CAP is no better than 3-competitive for $v \leq \min\{\frac{1+\rho}{2\rho}, \frac{1-\rho}{8\rho}\}$ (resp. $v \leq \min\{\frac{1}{3}, \frac{1-\rho}{6\rho}\}$).*

*Proof.* We prove this result using an input sequence $\mathcal{I}$ consisting of two streams of intruders, one stream each arriving at opposite endpoints. Specifically, at location 1, one intruder arrives at every time instant $\frac{2\rho(2i+1)k}{i}$, $0 \leq k \leq K$ for some very large $K$. At location $-1$, one intruder arrives at every time instant $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv} + \frac{k2\rho}{i}$, $0 \leq k \leq K$ for the same $K$.

As the first intruder at location 1 arrives $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv}$ time units before the first intruder arrives at location $-1$, CAP moves the defender to $\rho$ from the origin, reaching $\rho$ at time $3\rho$. As the intruders at location 1 arrive every $2\rho(2i+1)/i$ time units and the intruders at location $-1$ arrive every $2\rho/i$ time units, from this moment on until the end, the defender remains at $\rho$ because there will always be at most one intruder in $S^{j+\lfloor i+1 \rfloor}_{\text{opp}}$ and exactly one intruder in $S^{j+1}_{\text{same}}$, $S^{j+2}_{\text{same}}$ until $S^{j+2i+1}_{\text{same}}$ combined. Thus, except at the beginning and at the end of the streams, every $2\rho(2i+1)/i$ time units, the defender captures 1 intruder that arrives at $+1$ but loses $2i+1$ intruders that arrive at $-1$.

For $v \leq \frac{i(1-\rho)}{2\rho(2i+1)}$, a simple optimal offline algorithm that moves the defender to $-\rho$ and captures all the intruders that arrive at location $-1$ captures $\frac{2i+1}{2i+2}$ of the intruders. Thus, for the specified parameter settings, CAP cannot be better than $2i+1$-competitive.

We now consider the case when $i = 0.5$ (resp. $i = 1$) and show that an optimal offline algorithm $\mathcal{O}$ can capture all intruders for specific parameter settings. First consider $\rho + \frac{\rho}{i} \leq 1$. Let $p_1$ and $p_2$ denote two distinct points which are $\rho$ and $-\rho - \rho/i$ distance away from the origin respectively.

$\mathcal{O}$ moves the defender towards $p_1$ at time instant $\frac{1-\rho}{v} - \rho$ reaching $p_1$ just in time to capture the first intruder that arrived at endpoint 1. Since the first intruder at endpoint $-1$ arrives $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv}$ time after the arrival of first intruder at endpoint $+1$, the defender captures the second intruder at $p_2$ by moving immediately towards $p_2$. This is because the intruder will be located at a distance of $(2i+1)\rho v/i + \rho/i$ from $-\rho$ at the time when the defender captured the intruder at $p_1$. This means that the distance between the defender and the intruder will be $\frac{(2i+1)\rho(1+v)}{i}$ implying that the defender captures this intruder in $2\rho + \frac{\rho}{i}$ time or equivalently at location $p_2$. The defender takes $2(2i+1)\rho/i$ time units to move from $p_1$ to $p_2$ and back to $p_1$. Thus, the defender can capture every intruder, that arrives at endpoint 1 at time instant $(2i+1)2k\rho/i$, at location $p_1$ at time instant $\frac{1-\rho}{v} + (2i+1)2k\rho/i$.

Now consider the intruders that arrive at $-1$. They arrive $2\rho/i$ time units apart, and we already showed that the first intruder that arrived at $-1$ can be captured at $p_2$. This means that from the moment the defender leaves $p_2$ after capturing an intruder, the next intruder will take $2\rho/i$ time to reach $p_2$ and $\frac{2\rho}{i} + \frac{\rho}{iv}$ time units to reach $-\rho$, whereas the defender will take $4\rho + \frac{\rho}{i}$ time to move from $p_2$ to $p_1$ and then to $-\rho$. Thus, in order to ensure that the next intruder is not lost while the defender moves from $p_2$ to $p_1$ and then to $-\rho$, we need $4\rho + \frac{\rho}{i} \leq \frac{2\rho}{i} + \frac{\rho}{iv}$ implying $v \leq \frac{1}{4i-1}$, which always holds for $i = 0.5$ and yields $v \leq \frac{1}{3}$ for $i = 1$. In summary, after capturing an intruder at $p_1$, the defender moves to $p_2$, capturing an intruder that arrives at $-1$ at time $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv} + \frac{2\rho k(2i+1)}{i}$ at location $p_2$.

We now consider $\rho + \frac{\rho}{i} > 1$. Since $\rho + \frac{\rho}{i} > 1$, we cannot set the point $p_2$ at a distance of $-\rho - \frac{\rho}{i}$; instead, we set $p_2$ to be $-1$ and have the defender idle at $-1$ for $(2\rho - 2) + \frac{2\rho}{i}$ time. Note that the total time that the defender takes to move from $p_1$ to $-1$ combined with the waiting time at $-1$ is $3\rho - 1 + \frac{2\rho}{i}$. This time is sufficient for the defender to capture the intruders that arrives at time $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv} + \frac{2k\rho(2i+1)}{i}$ immediately. The next intruder, however, arrives at time $\frac{(2i+1)\rho}{i} + \frac{\rho}{iv} + \frac{2k\rho(2i+1)}{i} + \frac{2\rho}{i}$ which is after the defender leaves $-1$ as the defender leaves $-1$ at time $3\rho - 1 + \frac{2\rho}{i} + \frac{2k\rho(2i+1)}{i}$. This is equivalent to the intruder arriving at $-1$, $1 + \frac{\rho}{iv} + \frac{\rho(2i+1)}{i} - 3\rho$ time after the defender has left $-1$. The total time the defender takes to move from $-1$ to $\rho$ and then to $-\rho$ is $1 + 3\rho$ and the intruder will take $1 + \frac{\rho}{iv} + \frac{1-\rho}{v} + \frac{\rho(2i+1)}{i} - 3\rho$ time. Thus, in order to ensure that the next intruder is not lost, we need $3\rho \leq \frac{\rho}{iv} + \frac{1-\rho}{v} + \frac{\rho(2i+1)}{i} - 3\rho$. Substituting $i = 0.5$ yields $3\rho \leq \frac{2\rho}{v} + \frac{1-\rho}{v} + \rho$, which implies $v \leq \frac{1+\rho}{2\rho}$ and substituting $i = 1$ yields $v \leq \frac{1}{3\rho}$. As, for $i = 1$, $v \leq \frac{1}{3}$ holds, then this implies that $v \leq \frac{1}{3\rho}$ always holds. This concludes the proof. □

**Remark 2** (Evasive motion of intruders)**.** *We say that an intruder* evades *when the intruder moves toward the endpoint that generated the intruder or remains stationary. Since the analysis in Theorems 5, 6 and the Sweep algorithm are independent of the nature of the motion of intruders, the two fundamental results as well as Theorem 8 hold for evasive intruders. Furthermore, by modifying the set $S^k_{\text{opp}}$ in Algorithm 1 to $[-1, -(\rho + 2v\rho)]$ (resp. $[1, (\rho + 2v\rho)]$), it can be shown that 1 is 2-competitive for evasive intruders as well.*

### 2.3 Escaping Intruders

This section addresses problem **P2** with the physical region being $\mathcal{R}(\rho) = [-\rho, \rho]$ with $\rho > 1$, where intruders arrive at either $-1$ or $1$ and they move at a fixed speed $0 < v < 1$ towards $-\rho$ and $\rho$, respectively, as depicted in Figure 2.1b. Recall that if they pass their destination before the defender, moving at speed 1, reaches them, they are not captured. If the defender reaches them before they pass the target point ($-\rho$ or $\rho$), then they are captured.

#### 2.3.1 Fundamental Limits

Similar to Section 2.2, we start by deriving a fundamental limit for $c$-competitiveness of any algorithm.

**Theorem 15** (Necessary condition for $c$-competitiveness)**.** *For any problem instance $\mathcal{P}$ with $v > \frac{\rho-1}{\rho+1}$, there does not exist a $c$-competitive algorithm.*

*Proof.* Assume that the defender starts at the origin for both online algorithm and the optimal offline algorithm. Similar to the proof of Theorem 5, we start by constructing an input sequence $\mathcal{I}$. The input sequence consists of two stages: a stream of intruders that arrive at location 1, $\max\{2, \frac{\rho-1}{v}\}$ time units apart starting at time 1 and a burst of $c+1$ intruders which arrive at location $-1$ at time $t$. Time instant $t$ is the time instant the defender moves to 1 according to any online algorithm for the first time. Note that if the defender never moves to location 1, the stream never ends, and the burst never arrives, so the algorithm will not be $c$-competitive for any constant $c$, and the result follows.

Let $i$ be the number of stream intruders released up to and including time $t$; note that $i$ will be at least 1 if the defender reaches 1 exactly at time 1 when the first stream intruder is released. We first observe that the defender will capture at most intruder $i$. In particular, because the stream intruders are released $\max\{2, \frac{\rho-1}{v}\}$ time units apart, all stream intruders before $i$ have reached $\rho$ before time $t$. Two cases arise:

*Case 1*: If $t = 1$, this means the defender reached location 1 exactly at time 1 capturing the intruder that arrived at location 1 at time instant 1. Since $v > \frac{1-\rho}{1+\rho}$, the defender will not be able to capture the burst of $c+1$ intruders that arrive at time $t = 1$. We now consider how many intruders the optimal offline algorithm could have captured. For the same input instance as for the online algorithm, the optimal offline algorithm can move the defender to location $-1$ by time $t$ and will thus capture all $c+1$ burst intruders, and the result follows.

*Case 2*: If $t > 1$, this means the defender reached location 1 after time 1. In this case, the optimal offline algorithm can capture the first $i-1$ stream intruders immediately upon arrival by moving the defender to location 1 at time 1 and staying there until the first $i-1$ stream intruders have been captured. This algorithm can then move the defender to position $-1$ before the burst intruders have been released since the stream intruders arrive $\max\{2, \frac{\rho-1}{v}\}$ time units apart. Note that if $i = 1$, the algorithm moves the defender immediately to $-1$.

In either case, we show that the optimal offline algorithm can capture all the burst intruders and at least all but the last stream intruder whereas the online algorithm will capture at most 1 stream intruder, and the result follows. □

**Corollary 3.** *For any problem instance $\mathcal{P}$ with $v > \frac{\rho-1}{\rho+1}$, no algorithm (online or offline) can capture all intruders.*

*Proof.* Consider the same input instance as described in the proof of Theorem 15. The result then follows by observing there are choices for $t$ including $t = 1$ such that no algorithm can capture all $i$ stream intruders and all $c+1$ burst intruders, for same $t$ and $i$ as defined in the proof of Theorem 15. This concludes the proof. □

### 2.3.2 Algorithms

We now analyze two algorithms for the escaping intruders scenario. Note that CAP is not effective for escaping intruders since the perimeter locations $-\rho$ and $\rho$ are far apart.

#### 2.3.2.1 Sweeping Algorithm

Sweep is fundamentally the same as described for incoming intruders in Section 2.2; the only change is that the endpoints are now $-\rho$ and $\rho$ rather than $-1$ and 1. That is, the defender starts at the origin and moves toward $\rho$. Thereafter, the defender does not change speed or direction until it reaches an endpoint in which case it moves at unit speed towards the other endpoint.

**Theorem 16.** *Sweep is 1-competitive for any problem instance $\mathcal{P}$ for $v \leq \frac{\rho-1}{1+3\rho}$. Otherwise, Sweep is not c-competitive for any constant c.*

*Proof.* Similar to the proof of Theorem 8 assume that $v \leq \frac{\rho-1}{1+3\rho}$. Any intruder that arrives in the physical region $\mathcal{R}(\rho)$ takes $\frac{\rho-1}{v}$ time to reach the perimeter after arriving in $\mathcal{R}(\rho)$. Without loss of generality, consider an arbitrary intruder $i$ that arrives at location 1. In the worst case for Sweep, which is that intruder $i$ arrived just after the defender left location 1, the defender will take $3\rho + 1$ time to reach $\rho$. This follows as the defender first moves to location $-1$ from $\rho$ and then moves in the direction of the intruder, i.e., towards $\rho$. Since $3\rho+1 \leq \frac{\rho-1}{v}$ holds, then Sweep's defender will get to $\rho$ first and intruder $i$ will be captured, and the first result follows.

Now suppose that $v > \frac{\rho-1}{1+3\rho}$. Then there is an input instance where intruders only arrive at location 1 just after Sweep's defender has left location $+1$. Given that $v > \frac{\rho-1}{1+3\rho}$, all intruders will be lost and the second result follows. □

Next we describe a 4-competitive algorithm. For this algorithm, we make an assumption on the finiteness of the number of intruders that can arrive in the physical region $\mathcal{R}(\rho)$. This is because, unlike the previous algorithms, the defender now keeps an account of the intruders at every time instant. Thus, if the arrival of the intruders does not end in finite duration, it is possible that the defender does not capture any intruder and so the algorithm will not be competitive for any constant $c$.

#### 2.3.2.2 Instantaneous Compare and Capture Algorithm

Earlier, for incoming intruders, Algorithm CaC could compare the intruders that arrived in a particular time interval $T$ from each entry point, capture the intruders that arrived in $T$ from one entry point, and still be able to capture any intruders that arrived after $T$. In the escaping intruders case, this is not possible. Specifically, the defender may not be able to capture its currently targeted intruders and still capture intruders that arrived at the other arrival location after making its current capture plan. Thus, Instantaneous Compare and Capture algorithm (iCaC) constantly monitors for new arrivals and considers changing direction. We make the following assumption:

**Assumption A1:** Finite number of intruders arrive in the physical region $\mathcal{R}(\rho)$.

Note that this assumption implies that the last intruder arrives at a finite time instant. To determine if the defender should start chasing intruders at a time instant $k$, iCaC tracks three quantities: the total number of intruders it has captured $n_c^k$ and lost $n_l^k$ until instant $k$, as well as the total number of intruders that exist in the physical region $n_e^k$. iCaC tracks these quantities from the most recent time instant these quantities were set to zero which may be time 0. iCaC should begin a chase at time $k$ if the *chase condition* **C1**, defined as $2n_c^k < n_l^k + n_e^k$, holds. While the defender is chasing intruders at some time instant $k$, iCaC monitors the balance of intruders on the left and right; more precisely, it tracks two sets of intruders: $S_{\text{left}}^{k_l}$ which is the set of intruders that arrived at location $-1$ no earlier than time instant $k_l \leq k$ and $S_{\text{right}}^{k_r}$ which is the set of intruders that arrived at location 1 no earlier than time instant $k_r \leq k$. The time instants $k_l$ and $k_r$ might not be equal in order to track the intruders in $S_{\text{left}}^{k_l}$ and $S_{\text{right}}^{k_r}$ appropriately. We will formally define $k_l$ and $k_r$ shortly. In iCaC, the defender either chases intruders in $S_{\text{left}}^{k_l}$ or in $S_{\text{right}}^{k_r}$, and it switches direction when the other set of intruders is at least double the current set of intruders it is chasing. We describe this more precisely when we define the two *chase modes*.

A key parameter for iCaC is the *capture limit*; the point of no return such that if the defender crosses it, then it is impossible for the defender to capture intruders escaping on the other side. Some problem instances do not have a capture limit meaning the sweeping algorithm is guaranteed to capture all intruders.

**Definition 17** (Capture limit)**.** *If $\frac{\rho(1-v)-1-v}{2v} < \rho$, then the physical region $\mathcal{R}(\rho)$ for $\rho > 1$ has two capture limits, namely $\frac{\rho(1-v)-1-v}{2v}$ and $-\frac{\rho(1-v)-1-v}{2v}$, which we denote as $p_c$ and $-p_c$, respectively.*

We outline the iCaC algorithm in Algorithm 3 and describe the three modes, namely rest mode, chase mode and locked mode, that iCaC operates in, below.

*Rest mode:* The defender is located at either 1 or $-1$ and **C**1 does not hold. This means that the defender has captured sufficiently many intruders and can stay at the same location at time instant $k$. The defender resets $n_c^k$, $n_e^k$ and $n_l^k$ to zero (line 9-11 in Algorithm 3). With the reset of $n_c^k$, $n_e^k$ and $n_l^k$ to zero, a single intruder arriving on the opposite side is sufficient to make the defender leave rest mode thus guaranteeing that iCaC will not rest on many early captures to allow many later arriving intruders to escape. iCaC exits rest mode and enters either chase mode (line 5) or locked mode if **C1** holds.

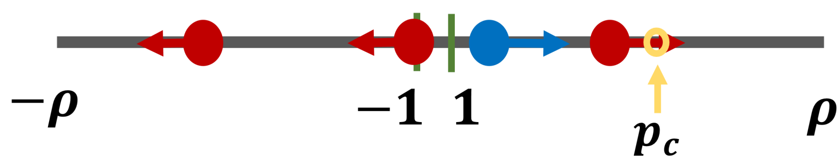



Figure 2.4 Snapshot at time instant $k$ for $p_c > 1$ when the defender switches from chase-right mode to chase-left mode as **C2** holds.

*Chase mode:* We have two modes under *chase mode*, *chase-right* and *chase-left*, where the defender is chasing intruders but can change direction instantaneously; these are the most complex modes and represent the key new modes for escaping intruders. iCaC can be in either of the two chase modes when the defender is in the range $[-p_c, p_c]$. In particular, when $p_c \geq 1$, iCaC is in chase-left (resp. chase-right) mode when the defender is in the range $[-p_c, p_c]$ and is moving towards $-\rho$ (resp. $\rho$). When $p_c < 1$, iCaC is in chase-left (resp. chase-right) mode when the defender is located in $[-p_c, 1)$ (resp. $(-1, p_c]$) and is moving towards $-\rho$ (resp. $\rho$). For $p_c \geq 1$, we define the chase region as the range $[-p_c, p_c]$ and for $p_c < 1$, we define chase region as the range $[-p_c, 1)$ (resp. $(-1, p_c]$) when the defender is in chase-left (resp. chase-right) mode. During chase mode, the defender updates $n_c^k$, $n_l^k$ and $n_e^k$ at every time instant.

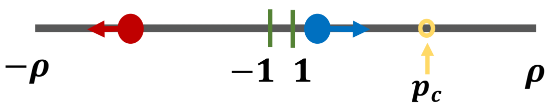
iCaC monitors a *switch condition* **C2** during either of the two chase modes at every time instant. In the chase-right mode, **C2** corresponds to $|S_{\text{left}}^{k_l}| \geq 2|S_{\text{right}}^{k_r}|$. In the chase-left mode, **C2** corresponds to $|S_{\text{right}}^{k_r}| \geq 2|S_{\text{left}}^{k_l}|$. As the name implies, if the corresponding switch condition applies, iCaC toggles its chase mode (lines 14-17) and reverses defender direction (cf Fig: 2.4). The switch condition ensures that the payoff for changing direction is large enough to warrant the cost of switching directions. Otherwise, the defender continues to move in the same direction (lines 18-20). If the defender captures all intruders it was chasing before reaching the capture limit, the defender enters locked mode (defined later) where it will return directly to 1 or $-1$.

We now define how Algorithm iCaC sets time instants $k_l$ and $k_r$ and updates $n_e^k$, $n_c^k$, and $n_l^k$ when a switch occurs. When Algorithm iCaC first enters a chase mode from either rest mode or locked mode, assuming there are a finite number of intruders, it will go through

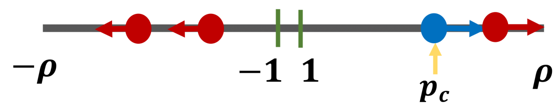
**Algorithm 3:** Instantaneous Compare-and-Capture Algorithm

Defender is at origin
**for** each time instant $k$ **do**
  Update $n_e^k$, $n_c^k$, $n_l^k$.
  **if** defender is in rest mode **then**
    **if** *chase condition holds* **then**
      Set $k_l = k_r = t_1$.
      Enter chase-left (resp. chase-right) mode. Move to capture intruders that arrive at $-1$ (resp. 1).
    **else**
      Stay at 1 (resp. $-1$) and reset $n_e^k$, $n_c^k$, $n_l^k$ to 0.
    **end**
  **else if** *defender is in chase-right (resp. chase-left) mode* **then**
    **if** *switch condition holds* **then**
      Add $|S_{\text{right}}^{k_r}|$ (resp. $|S_{\text{left}}^{k_l}|$) to $n_l^k$. Set $k_r = k$ (resp. $k_l = k$).
      Move to capture $|S_{\text{left}}^{k_l}|$ (resp. $|S_{\text{right}}^{k_r}|$).
    **else**
      Move to capture $|S_{\text{right}}^{k_r}|$ (resp. $|S_{\text{left}}^{k_l}|$).
    **end**
    **if** $|S_{\text{right}}^{k_r}|$ *(resp.* $|S_{\text{left}}^{k_l}|$*) is captured or defender exits chase region* **then**
      Enter locked mode and reset $n_e^k$, $n_c^k$, $n_l^k$ to 0.
    **end**
  **else**
    **if** *defender is located in chase region* **then**
      Return to nearest location out of 1 or $-1$.
    **else**
      Move to capture all intruders on same side.
      Return to nearest location out of 1 or $-1$ after capture.
    **end**
    **if** *chase condition holds after return to* 1 *or* $-1$ **then**
      Set $k_l = k_r = t_{e-1}^l$ and enter chase mode.
    **else**
      Enter rest mode reset $n_e^k$, $n_c^k$, $n_l^k$ to 0.
    **end**
  **end**
**end**

a sequence of $n$ consecutive chase modes for some $n \geq 1$ before ending in locked mode. We assume without loss of generality that the first chase mode is chase-right. The defender begins this first chase mode at position $-1$ at some time $t_1$. For this first chase mode, $k_l = k_r$ and are set equal to $t_1$ if the previous mode was rest mode or $t_1 = 0$. Otherwise, $k_l = k_r$



(a) Defender is in chase region and has captured intruders it was pursuing at time instant $k$.



(b) Defender is at location $p_c$ and **C2** does not hold.

Figure 2.5 Snapshots at time instant $k$ when defender enters locked mode for $p_c > 1$.

and are set equal to the start of the previous locked mode. For the $i$th chase mode in this sequence for $2 \leq i \leq n$, we define $t_i$ to be the time when the switch condition caused iCaC to change direction for the $(i-1)^{th}$ time. If $i$ is even, then $k_l = t_{i-1}$ and $k_r = t_i$; if $i$ is odd, then $k_r = t_i - 1$ and $k_l = t_i$. This assignment of $k_l$ and $k_r$ is required to ensure an accurate comparison of intruders on either side for the switch condition. In either case, Algorithm iCaC adds the intruders that the defender was pursuing prior to reversing the direction to $n_l^k$ i.e., the defender will not include these intruders in either $S_{\text{left}}^{k_l}$ or $S_{\text{right}}^{k_r}$ again and will not count such intruders as captured even if the defender does capture them.

*Locked mode:* In this mode, the defender's route is fixed until it returns to the nearest location out of 1 or $-1$. At the start of locked mode, Algorithm iCaC resets $n_c^k$, $n_e^k$, and $n_l^k$ to 0 and then updates these quantities as appropriate but will not change mode until it returns back to $-1$ or $+1$ after capturing intruders. When the defender returns to $-1$ or $+1$, **C1** will dictate whether iCaC enters rest mode or one of the chase modes. If the defender is in the chase region, then the defender enters locked mode only when the defender has captured all intruders it is pursuing (lines 21-24, cf Fig. 2.5a). In this case, the defender returns to the nearest location out of 1 or $-1$ (lines 26-28). For $p_c > 1$, the defender is guaranteed to be in locked mode if it is in the region $[-\rho, -p_c) \cup (p_c, \rho]$. In particular, Algorithm iCaC transitions from chase-left mode (resp. chase-right mode) to locked mode at time $k$ if the defender is at $-p_c$ (resp. $p_c$) and **C2** does not hold (cf Fig. 2.5b). In this case, the defender's route is locked into capturing all the intruders it can that are to its left (resp. right). Once it has done so, it returns to $-1$ (resp. 1) (lines 29-31). Recall that for $p_c < 1$, Algorithm iCaC is in chase-left (resp. chase-right) mode in $[-p_c, 1)$ (resp. $(-1, p_c]$). So, Algorithm iCaC is in locked mode when the defender is in $[-\rho, -p_c) \cup [1, \rho]$ (resp. $[-\rho, -1] \cup (p_c, \rho]$). In this case, the defender crosses $-1$ (resp. 1) when pursuing intruders after entering locked mode from chase-left (resp. chase-right) mode and then returns to $-1$ (resp. 1) after capture.

We now prove two properties of algorithm iCaC.

**Lemma 18.** *If the defender leaves* 1 *(resp.* $-1$*) at time* $t$ *and goes directly to* $-p_c$ *(resp.* $p_c$*) and then turns around and goes to* $\rho$ *(resp.* $-\rho$*), it will arrive at* $\rho$ *(resp.* $-\rho$*) before any intruder that arrives at* 1 *(resp.* $-1$*) strictly after time* $t$*.*

*Proof.* The time it takes for the defender to leave 1 and go to $-p_c$ and then go to $\rho$ is exactly $2p_c + 1 + \rho$. The time it would take an intruder to go from 1 directly to $\rho$ is $\frac{\rho-1}{v}$. Since the defender will leave 1 before the intruder in question arrives, we need to show that $2p_c + 1 + \rho \leq \frac{\rho-1}{v}$ which is true given our definition of $p_c$. This concludes the proof. □

Note that if $p_c \geq \rho$, then Sweep is guaranteed to capture all intruders and iCaC is not necessary, so we assume that $p_c < \rho$ for the remainder of this algorithm description.

In order for Algorithm iCaC to be effective, we assume that $v \leq \frac{3(\rho-1)}{7\rho+1}$. This ensures that if the defender crosses either capture limit at time $t$, it has sufficient time to capture any intruders it was pursuing plus any intruders that arrive after time $t$. We formalize this condition below.

**Lemma 19.** *Let the defender be in the locked mode at time* $t$*. Then, the defender can capture all intruders that arrive in the physical region* $\mathcal{R}(\rho)$ *at time* $t' \geq t$ *for* $v \leq \frac{3(\rho-1)}{7\rho+1}$*.*

*Proof.* We first consider the case when the defender is in $[-\rho, -p_c) \cup (p_c, \rho]$. In the worst case, the time it takes the defender to leave $p_c$ (resp. $-p_c$) and go to $\rho$ (resp. $-\rho$) and then go to $-\rho$ (resp. $\rho$) is exactly $3\rho - p_c$. The time it would take an intruder to go from $-1$ (resp. 1) directly to $-\rho$ (resp. $\rho$) is $\frac{\rho-1}{v}$. Thus, we simply need to show that $3\rho - p_c \leq \frac{\rho-1}{v}$ which is true given $v \leq \frac{3(\rho-1)}{7\rho+1}$. For the case when the defender enters the locked mode in $[-p_c, p_c]$, Lemma 18 ensures the capture of the intruders. This concludes the proof. □

A consequence of Lemma 19 is that there is no benefit for intruders to arrive at either location 1 or $-1$ once the defender enters the locked mode. We now establish that Algorithm 3 is 4-competitive.

**Theorem 20.** *For input instances with a finite number of intruders, the Instantaneous Compare and Capture (iCaC) algorithm defined in Algorithm 3 is* 4*-competitive for any problem instance* $\mathcal{P}$ *for which* $v \leq \frac{3(\rho-1)}{7\rho+1}$.

*Proof.* We first observe that Algorithm iCaC would not be competitive against some input instances with an infinite number of intruders as Algorithm iCaC can be forced to keep changing directions without capturing any intruders. Our assumption **A1** of a finite number of intruders implies that the last intruder must arrive at a finite time instant $t$. If the chase condition never holds which means the defender is always in rest mode, it is obvious that $n_c \geq n_l/2$. We thus assume for the rest of the proof that the chase condition holds at least once.

We start by defining an epoch $\tau_e$ which starts at the time instant $t_e$ when the defender leaves locked mode and ends at the time instant $t_{e+1}$ when the defender leaves the next consecutive locked mode. The first epoch $\tau_1$ starts at time 0, i.e., we treat time 0 as the end of locked mode. As the number of intruders is finite, the defender will be in locked mode at least once, so one epoch must exist.

In any epoch $\tau_e$, the defender will enter a sequence of $n$ chase modes for some finite $n \geq 1$, starting at times $t_e^{c,i}$ for $1 \leq i \leq n$, and then locked mode, starting at time $t_e^l$. Recall that, for condition **C2**, Algorithm iCaC sets $k_l$ and $k_r$ to the time instances $t_e^{c,i}$ depending on the type of chase mode. The defender may skip rest mode completely meaning $t_e = t_e^{c,1}$, or the defender starts in rest mode at $t_e$ before transitioning to chase mode at $t_e^{c,1} > t_e$, i.e., the defender is in rest mode in the interval $[t_e, t_e^{c,1})$.

During epoch $\tau_e$, we account for all the intruders that arrived from $t_{e-1}^l$ (the start of the previous epoch's locked mode) or time 0 for $\tau_1$ to time $t_e^l$. Suppose Algorithm iCaC enters rest mode at time $t_e$. This means the defender caught at least 1/3 of the intruders that arrived during the interval $[t^l_{e-1}, t_e]$ so the chase condition does not hold at time $t_e$. Likewise, during rest mode, the defender resets $n^k_c$, $n^k_l$ and $n^k_e$ to zero until time $t^{c,1}_e$ when the number of intruders that arrive at the other entry point are more than double the number of intruders that arrive at the entry point the defender is at. Thus, in summary, Algorithm iCaC captures at least 1/3 of all the intruders that arrived in the time interval $[t^l_{e-1}, t^{c,1}_e)$. We then only need to account for the intruders that arrive from $[t^{c,1}_e, t^l_e)$ when analyzing the sequence of chase modes and locked mode. On the other hand, if Algorithm iCaC skips rest mode, then we must account for all the intruders that arrived since $t^l_{e-1}$ when analyzing the sequence of chase modes and locked mode.

We now analyze the sequence of $n$ chase modes, $n \geq 1$. One crucial observation is that because of Lemma 18 and Lemma 19 and the fact that Algorithm iCaC adds any intruders it was chasing before switching directions to $n^k_l$, all the intruders the defender is chasing during a given chase mode will be captured if the defender does not switch directions again in the chase mode. We focus on the final chase mode which starts at time $t^{c,n}_e$. Without loss of generality, we assume this $n$th chase mode is chase-right mode. Let $a_n$ be the number of intruders in $S^{k_r}_{\text{right}}$, $k_r = t^{c,n-1}_e$, at time $t^{c,n}_e$ that caused Algorithm iCaC to move towards $\rho$. Specifically, the intruders $a_n$ are all of the intruders that arrived at location 1 in the interval $[t^{c,n-1}_e, t^{c,n}_e]$. We now characterize the number of intruders lost that arrived before time $t^{c,n}_e$. These correspond to the number of intruders that were pursued but never captured in the earlier $n-1$ chase modes (which might be 0 if there were no such chase modes). Recall that Algorithm iCaC adds these intruders to $n^k_l$ after switching direction in the chase mode. Because the switch condition requires the number of intruders in the opposite direction to be twice the number of intruders in the direction the defender was pursuing, this number of intruders is at most $a_n/2 + a_n/4 + \cdots + a_n/2^{n-1}$ which is strictly less than $a_n$.

We now characterize the number of intruders $b_n$ that arrived during $[t^{c,n}_e, t^l_e]$ at entry point $-1$ and thus might be lost. Note that these $b_n$ intruders correspond to $|S^{k_l}_{\text{left}}|$ where $k_l = t^{c,n}_e$. Let $a'_n$ be the number of intruders that arrived at entry point 1 that were captured as well

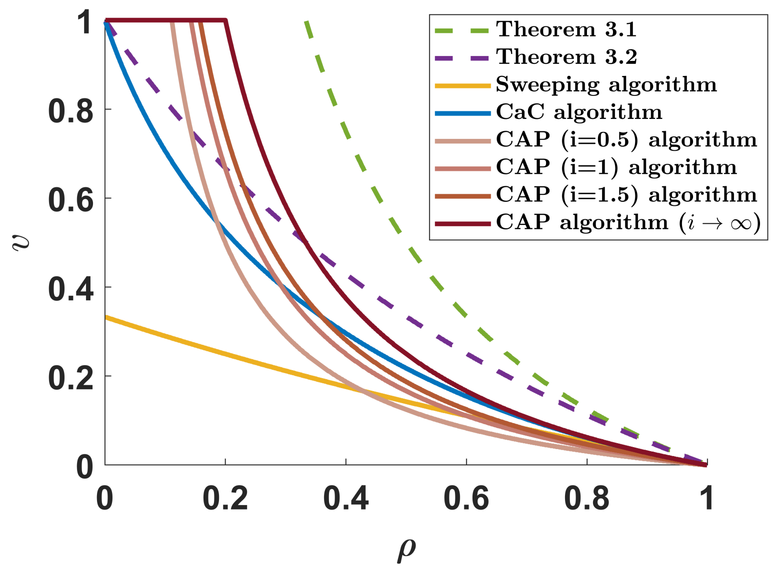



Figure 2.6 Parameter regimes for incoming intruders (**P1**). Solid lines, extend to the left and lower bounds (dashed lines), extend to the right.

as pursued during $[t_e^{c,n}, t_e^l]$ which might be larger than $a_n$ since new intruders might have arrived at 1 after $t_e^{c,n}$. Because of the switch condition, we know that $b_n < 2a_n'$. Finally, from Lemma 18, Algorithm iCaC will capture all $a_n'$ intruders during the $n$th chase mode or the ensuing locked mode. This means that the number of intruders lost in epoch $\tau_e$, $a_n + b_n$, is strictly less than $3a_n'$, the number of intruders captured during the sequence of chase modes and locked modes of epoch $\tau_e$.

This analysis applies to all epochs meaning Algorithm iCaC captures at least 1/4 of all intruders that arrive before the stock of locked mode for the final epoch. The final epoch must end with a transition to rest mode meaning that iCaC captures at least 1/3 of the intruders that arrive after the start of locked mode for the final epoch. Thus, the result follows. □

We now provide two parameter regime plots highlighting the parameter regime in which our algorithms are effective as well as the parameter regime for the established necessary conditions for both problem variants **P1** and **P2**.

## 2.4 Results and Discussion

Figure 2.6 and Figure 2.7 show $v$-$\rho$ plots summarizing our results for incoming (**P1**) and escaping (**P2**) intruders, respectively.

For incoming intruders (Figure 2.6), we see that CAP has relatively small area of utility. This is the region between the curves for CAP and CaC which increases as the input $i$ to CAP increases. Note that CAP does not extend beyond the dark red curve ($i \to \infty$) in Figure 2.6. For $\rho \geq 0.9$, the parameter curve of CAP for $i \to \infty$ overlaps that of the CaC algorithm implying that for any $\rho < 0.9$, there exists a CAP algorithm whose parameter regime is completely above that of the CaC. Although CAP covers higher parameter regime for higher values of $i$, the increase in the parameter regime decreases as $i$ increases implying that for high values of $i$, slight increase in the parameter regime may increases the competitiveness of CAP considerably. Interestingly, CAP's regions extend beyond the curve for Theorem 6 and for small values of $\rho$, CAP's parameter regime covers all values of intruder speed $v$. Finally, we observe that for $\rho > 0.95$, the parameter curve for CaC and CAP ($i \to \infty$) overlaps to that of the Sweep. Key open problems include closing the 2-competitive gap between the CaC curve and the curve for Theorem 6 and closing the $c$-competitive gap between the CAP ($i \to \infty$) curve and the curve for Theorem 5.

For escaping intruders, we plot the lower bound and the two algorithms. We observe that the region of utility for Sweep is completely contained within the region of utility for Algorithm iCaC. The key open problem is to close the gap between the curves for Theorem 15 and Algorithm iCaC.

We now briefly comment on the choice of algorithms and the applicability of this work. The choice of which algorithm to use depends on the problem parameters, i.e., the size of the perimeter and the speed ratio, and the acceptable bound on competitiveness. Note that the acceptable competitive ratio is governed by the application. For instance, given the problem parameters, a military application may require algorithms with the least competitive ratio[1]

[1]Designing a 1-competitive algorithm may not always be feasible.

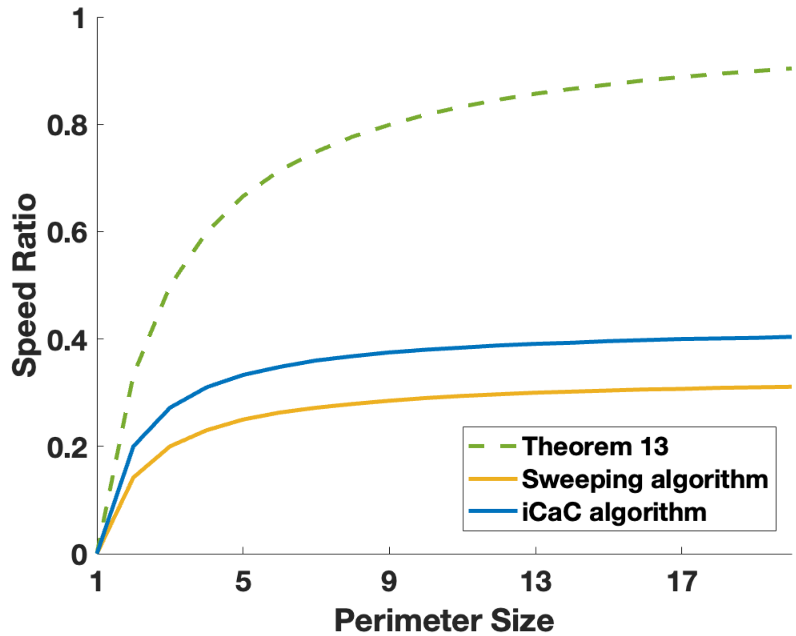



Figure 2.7 Parameter regimes for escaping intruders (**P2**). Solid lines, extend to the right and lower bounds (dashed lines), extend to the left.

whereas an application focused on relaying information may not have this requirement. Finally, we remark that the sufficient conditions on the algorithms are derived in the *worst-case*. In some applications, these worst-case inputs may not be realizable and thus the algorithms will perform better for those applications.

## 2.5 Conclusions

We study a problem where a single defender defends a linear perimeter from mobile intruders. A key contribution is the combination of concepts and techniques from competitive analysis of online algorithms with pursuit of multiple mobile intruders. We considered two scenarios: incoming intruders and escaping intruders. We designed and analyzed three classes of algorithms for incoming intruders. The first two algorithms are demonstrated to be 1, and 2-competitive, respectively. We generalized our third algorithm to a family of algorithms and demonstrated that it is $2i + 2$ competitive, where $i$ is an input for the generalized algorithm. For escaping intruders, we designed and analyzed two algorithms and proved that they are 1 and 4-competitive. We also derived fundamental limits on the $c$-competitiveness for any constant $c$ for both scenarios.

# CHAPTER 3

## RANDOMIZED COMPETITIVE PERIMETER DEFENSE IN LINEAR ENVIRONMENTS

In Chapter 2, our main focus was on the design and analysis of deterministic algorithms with provable competitive ratios. Recall that the inputs designed by the adversary heavily rely on the information about the online algorithms. In simpler terms, the adversary takes advantage of the information asymmetry to construct worst-case inputs against an online algorithm, prompting the question, "How can online algorithms limit the information available to the adversary?".

To address the aforementioned question, in this chapter, we design *randomized* algorithms to withhold some information from the adversary. Specifically, we address a problem in which a robotic defender, having a maximum unit speed, is tasked with defending a given set of points (perimeter) from mobile intruders. The intruders are released in an *arbitrary manner* at the endpoints of the line segment and move inwards, i.e., towards the midpoint of the environment. Upon release, each intruder moves with a fixed speed $v < 1$ towards the perimeter. The defender seeks to intercept maximum number of intruders before the intruders reach the perimeter. We focus on design and analysis of randomized *online* algorithms to route the defender as the input, consisting of release times, number of intruders, and release location of every intruder, is unknown a priori. The contributions of this chapter are as follows:

1. **Necessary conditions:** We establish necessary conditions on the competitiveness of any randomized online algorithm. Specifically, we characterize a parameter regime in the $v-\rho$ space in which no randomized algorithm can be better than 2-competitive, and a second parameter regime in which no algorithm is better than 1.33-competitive. We also characterize a parameter regime on the existence of 2-competitive algorithms[1] for a special class of randomized online algorithms that do not allow the vehicle to

[1] in the limit when an adversary can release arbitrarily many intruders

remain stationary. Such algorithms are called *zealous algorithms*.

2. **Algorithm design and analysis:** We design and analyze three randomized algorithms and establish their competitive ratios. Specifically, our first algorithm is provably 2-competitive for any value of $v$ and $\rho$ while the competitive ratio of our second and third algorithms depends on the problem parameters.

This chapter is organized as follows. In Section 3.1, we formally define our problem. Section 3.2 establishes the fundamental limits of the problem and Section 3.3 presents the randomized algorithms and their analysis. Section 3.4 presents and discusses the parameter regime plots and finally, Section 3.5 summarizes this chapter.

### 3.1 Problem Formulation

Analogous to the environment in Chapter 2, consider a linear environment $\mathcal{E} = [-1, 1]$ which consists of a region $\mathcal{R}(\rho) = \{y \in \mathbb{R} : |y| \leq \rho < 1\}$. The *perimeter* $\partial\mathcal{R}(\rho)$ is defined as the set of two locations $\{\rho, -\rho\}$, which needs to be defended from intruders. An adversary releases mobile intruders over time at either location $-1$ or 1. Each intruder moves with fixed speed $v < 1$ towards the nearest location in $\partial\mathcal{R}(\rho)$. The defense consists of a single vehicle with motion modeled as a single integrator. The vehicle is said to *capture* an intruder when the vehicle's location coincides with it. An intruder is said to be *lost* if the intruder reaches the perimeter without being captured. An intruder is said to be *capturable* if the vehicle can capture it before the intruder reaches the perimeter. Let $n(t)$ denote the number of intruders that arrive at time instant $t$ at endpoint $+1$. In this work we consider that $n(t) \leq 2N$, i.e., the adversary can release at most $2N, N \in \mathbb{Z}_{>0}$, intruders at each time instant. Further, we will also suppose that the adversary requires at least $\Delta \in \mathbb{R}_{>0}$ time to release intruders in the environment once it has released some intruders in the environment. More precisely, if $n(t) \leq 2N$ intruders are released at time instant $t$, then the next intruders are released no earlier than time $t + \Delta$. Note that if $\Delta \geq 2$ holds, then the problem is trivial as the defender can cover the environment before the next intruder arrives. Thus, we will focus on the case that $0 < \Delta < 2$. Finally, we denote the $n(t)$ intruders released at $+1$ at time $t$ as $i^n_{t,+1}$. Intruders released at endpoint $-1$ are denoted analogously. Since our problem is defined by four parameters, we denote a problem instance as $\mathcal{P}(\rho, v, N, \Delta)$ or $\mathcal{P}$ for brevity. With the $c$-competitive randomized algorithms defined in Chapter 1 (Definition 3), the objective of this chapter is the following.

**Problem Statement:** The aim of this chapter is to establish fundamental guarantees and design $c$-competitive randomized algorithms for the vehicle.

### 3.2 Fundamental Limits

We start with establishing the necessary conditions in the $(v, \rho)$ parameter space for any randomized algorithm. Our approach is based on *Yao's mini-max principle*, stated below, to lower bound the expected performance of any randomized algorithm.

**Lemma 21** (Yao's mini-max principle[45])**.** *Let $\bar{X}$ denote the probability distribution over a set of deterministic input instances $\mathcal{I}_x, x \in \mathcal{X}$. Suppose that there exists a constant $\underline{c} \geq 1$ such that*

$$\inf_{A_i \in \mathcal{A}_{det}} \{\mathbb{E}_{\bar{X}}[n_{\mathcal{O}}(\mathcal{I}_x)] - \underline{c}\mathbb{E}_{\bar{X}}[n_{A,i}]\} \geq 0,$$

*then the constant $\underline{c}$ is a lower bound on the competitive ratio of any randomized algorithm against an oblivious adversary, where $\mathcal{A}_{det}$ is the set of deterministic online algorithms as defined in Chapter 1.*

Yao's principle is a valuable tool that allows us to convert the supremum over deterministic inputs (cf. Definition 3) into a search using the expected performance over a finite number of inputs and then optimize the choice of the probabilities used to compute the expected values. We demonstrate this technique in establishing the fundamental limits for the problem. In particular, we first establish fundamental limits for a particular class of algorithms known as *zealous algorithms* and then establish fundamental limits for any randomized algorithms. We start with the definition of zealous algorithms.

**Definition 22.** *An algorithm $\mathcal{A}_z$ is called zealous if it satisfies the following:*

1. *If there are capturable intruders, then direction of vehicle as per $\mathcal{A}_z$ changes only if one of the conditions hold: i) a new intruder is released at an endpoint, ii) the vehicle is at the origin, or iii) the vehicle has just captured an intruder.*

2. *At any time if there are capturable intruders, $\mathcal{A}_z$'s vehicle moves towards a capturable intruder or the origin (if the vehicle is not at the origin).*

Note that in all of the proofs for the fundamental limits, we will start by describing a set of input sequences $\mathbf{I}$ and a probability distribution on $\mathbf{I}$. The set $\mathbf{I} = \{\mathcal{I}_\infty, \mathcal{I}_\in\}$, i.e., $\mathbf{I}$ consists of two input sequences $\mathcal{I}_\infty$ and $\mathcal{I}_\in$, each of which will be described in the proof. We then determine the performance of the optimal offline algorithm as well as the best deterministic algorithm and apply Yao's principle to establish the result.

**Theorem 23.** *Let $\epsilon := 1 + \rho - \frac{1-\rho}{v}$. For any problem instance $\mathcal{P}$, $\inf_{\mathcal{A}_z} c_{\mathcal{A}_z}(\mathcal{P}) \geq \frac{4N}{2N+1}$ if $v \geq \frac{1-\rho}{1+\rho}$ and*

$$\Delta < \begin{cases} \min\{1, 2\rho\}, & \text{if } v = \frac{1-\rho}{1+\rho} \\ \min\{1, \epsilon\}, & \text{otherwise.} \end{cases}$$

*Proof.* In this proof, the set of input sequences $\mathbf{I}$ consists of two input sequence, i.e., $\mathbf{I} = \{\mathcal{I}_1, \mathcal{I}_2\}$. Both input sequences in $\mathbf{I}$ start at time 0 and are described as follows. In input sequence $\mathcal{I}_1$, a single intruder is released at $+1$ at time 0. We denote this intruder as $i^0_{1,+1}$. At time 1, $2N$ intruders are released at endpoint $-1$ (denoted as $i^1_{2N,-1}$) and at time instant $t$, $2N$ intruders are released at $+1$ (denoted as $i^t_{2N,+1}$). The time instant $t = 1 + \Delta$ for $v = \frac{1-\rho}{1+\rho}$ and $t = 1+\epsilon$, otherwise. Similarly, in input sequence $\mathcal{I}_2$, a single intruder is released at $-1$ at time 0. At time 1, $2N$ intruders are released at endpoint $+1$ and at time instant $t$, $2N$ intruders are released at endpoint $-1$. Both $\mathcal{I}_1$ and $\mathcal{I}_2$ have equal probability of 1/2 of being selected. We now describe the performance of an optimal offline algorithm $\mathcal{O}$ followed by the performance of the best deterministic zealous algorithm $\mathcal{A}_z$. For both $\mathcal{O}$ and $\mathcal{A}_z$, we assume that the vehicle starts at the origin at time 0.

As $\mathcal{O}$ has the entire information of the selected input sequence, $\mathcal{O}$ can move the vehicle to the endpoint at which the first $2N$ intruders arrive. Suppose that $\mathcal{I}_1$ is selected. Then, $\mathcal{O}$ moves the vehicle towards $-1$ and captures $i_{1,-1}^{2N}$ immediately upon release. Upon capture, $\mathcal{O}$ then moves the vehicle directly towards $+\rho$, capturing the $2N$ intruders that were released at endpoint 1 at $\rho$. The capture of these intruders is ensured as these intruders arrive at time $t$. The performance of $\mathcal{O}$ against $\mathcal{I}_2$ is analogous. Thus, $\mathbb{E}[n_{\mathcal{O}}] = 4N$.

From zealousness of $\mathcal{A}_z$, it follows that the $\mathcal{A}_z$'s vehicle will move towards the single intruder that arrives at time 0. Further, after time 0, the vehicle can only change direction once it captures the single intruder. Suppose that input sequence $\mathcal{I}_1$ is selected. Then, since vehicle moves towards $+1$ at time 0, the vehicle will be $\frac{1}{1+v}$ distance away from the origin when it captures the first intruder. At this moment, the vehicle can either continue to move towards $+1$ or towards the origin.

Suppose that the vehicle moves towards $+1$ and consider that $v > \frac{1-\rho}{1+\rho}$ holds. Since $2N$ intruders arrive at $-1$, the vehicle will not be able to capture these intruders if $v > \frac{1-\rho}{1+\rho}$. Thus, in this case and against $\mathcal{I}_1$, the vehicle captures at most $2N+1$ intruders. Now consider that $v = \frac{1-\rho}{1+\rho}$ holds. Then, the vehicle can capture the $i_{2N,-1}^{1}$ intruders that arrive at endpoint $-1$ exactly at location $-\rho$ by moving towards $-\rho$ as soon as these intruders are released. Further, as $i_{2N,+1}^{t}$ intruders arrive at $+1$ at time $t = 1+\Delta$, the vehicle cannot capture these intruders since $\Delta < 2\rho$. Finally, if the vehicle changes direction when $i_{2N,+1}^{t}$ intruders are released, it cannot capture $i_{2N,+1}^{t}$ and $i_{2N,-1}^{1}$ as $v < 1$. Thus, against $\mathcal{I}_1$, the vehicle captures at most $2N+1$ intruders for $v \geq \frac{1-\rho}{1+\rho}$. The performance against $\mathcal{I}_2$ is analogous and has been omitted for brevity.

Now suppose that the vehicle decides to move towards the origin after capturing $i_{1,+1}^{0}$ against $\mathcal{I}_1$. In this case, it is guaranteed that the vehicle will not be at $-1$ by time 1 and thus, it follows that the vehicle will capture at most $2N+1$ intruders. As the performance against $\mathcal{I}_2$ is analogous, $\mathbb{E}[n_{\mathcal{A}_z}] = 2N + 1$. The result then follows from Lemma 21. $\square$

**Theorem 24** (Necessary condition for $c(\mathcal{P}) \geq 2$). *For any problem instance $\mathcal{P}$ for which $v > \frac{1-\rho}{\rho}$, $c(\mathcal{P}) \geq 2$ against an oblivious adversary.*

*Proof.* The set of input sequences $\mathbf{I}$ consists of two input sequences; $\mathcal{I}_1$ and $\mathcal{I}_2$. Both input sequences start at time instant $\rho$ and involves a single intruder $i$. Specifically, in input sequence $\mathcal{I}_1$, intruder $i$ arrives at endpoint $-1$ at time $t \geq \rho$. On the other hand, in input sequence $\mathcal{I}_2$ intruder $i$ arrives at endpoint $+1$ at the same time $t$. Both $\mathcal{I}_1$ and $\mathcal{I}_2$ have equal probability of 1/2 of being selected. Thus, our set of input sequences is $\mathbf{I} = \{\mathcal{I}_1, \mathcal{I}_2\}$ with a probability distribution $\bar{X}$ on $\mathbf{I}$. We now describe how the optimal offline algorithm and the best deterministic algorithm captures intruder $i$. We assume that the vehicle starts at the origin for both optimal offline and the deterministic algorithm.

Recall that once the input sequence $\mathcal{I}_s, s \in \{1, 2\}$ is selected, the optimal offline algorithm $\mathcal{O}$ has the entire information of the selected input sequence. As intruder $i$ arrives after time $\rho$ in both input sequence, $\mathcal{O}$ can move the vehicle in the direction of the endpoint where the intruder $i$ arrives. Given that the vehicle has covered at least $\rho$ distance moving towards the end point where $i$ arrives, capture of $i$ is ensured in both input sequences. This yields $\mathbb{E}_{\bar{X}}[n_{\mathcal{O}}] = \frac{1}{2} + \frac{1}{2} = 1$.

For any deterministic online algorithm $\mathcal{A}$, if the vehicle is at the origin at time $t$ then, since $\frac{1-\rho}{v} > \rho$, the vehicle cannot capture intruder $i$ from any input sequence in $\mathbf{I}$. Thus, the vehicle must be either located in $(0, 1]$ or $[-1, 0)$ by time $t$. Without loss of generality, assume that the vehicle is located in $(0, t]$ when intruder $i$ arrives. Intruder $i$ arrives at endpoint $+1$ with probability 1/2, in which case, $\mathcal{A}$ captures $i$. Otherwise, i.e., intruder $i$ arrives at endpoint $-1$ with probability $\frac{1}{2}$, the vehicle will not be able to capture intruder $i$ as $\frac{1-\rho}{v} > \rho$. Thus, $\mathbb{E}_{\bar{X}}[n_{\mathcal{A}}] = \frac{1}{2}$ and the result follows from Lemma 21. $\square$

The previous result establishes a necessary condition for the existence of online randomized algorithms with competitive ratio of at least 2. Then the next result establishes a necessary condition for the existence of online randomized algorithms with competitive ratio of at least 1.33 under specific parameter regime.

**Theorem 25** (Necessary condition for $c(\mathcal{P}) \geq 1.33$). *Let* $\epsilon := \begin{cases} 2\rho v, \ \textit{if } v = \frac{1-\rho}{1+\rho}. \\ 1+\rho - \frac{1-\rho}{v}, \ \textit{if } v > \frac{1-\rho}{1+\rho}. \end{cases}$. *Then, for any problem instance $\mathcal{P}$ that satisfies $v \geq \frac{1-\rho}{1+\rho}$ and $\Delta < \epsilon$, $c(\mathcal{P}) \geq \frac{4}{3}$ against an oblivious adversary.*

*Proof.* The set of input sequence consists of two input sequences; $\mathcal{I}_1$ and $\mathcal{I}_2$. In input sequence $\mathcal{I}_1$, a single intruder $i^1_{1,+1}$ arrives at endpoint $+1$ at time instant 1 and a single intruder $i^{1+\epsilon}_{1,-1}$ arrives at endpoint $-1$ at time $1+\epsilon$, where

$$\epsilon = \begin{cases} 2\rho v, \ \text{if } v = \frac{1-\rho}{1+\rho}. \\ 1+\rho - \frac{1-\rho}{v}, \ \text{if } v > \frac{1-\rho}{1+\rho}. \end{cases}$$

In input sequence $\mathcal{I}_2$, a single intruder $i^1_{1,-1}$ arrives at endpoint $-1$ at time instant 1 and a single intruder $i^{1+\epsilon}_{1,+1}$ arrives at endpoint $+1$ at time $1+\epsilon$, for the same $\epsilon$. Both input sequence have equal probability of 1/2 of being selected. Thus, we have set of input sequence $\mathcal{I} = \{I_x : x \in \{1,2\}\}$, with a probability distribution $\bar{X}$ on $\mathcal{I}$.

We first describe how the optimal offline algorithm $\mathcal{O}$ captures both intruders. As $\mathcal{O}$ knows the entire input sequence at time 0, $\mathcal{O}$ can have the vehicle move to endpoint $+1$ arriving endpoint $+1$ at exactly time $+1$, if input sequence $\mathcal{I}_1$ is selected. Otherwise, $\mathcal{O}$ can move the vehicle to endpoint $-1$, arriving endpoint $-1$ at exactly time 1. Either way, the vehicle captures the intruder that arrives at the endpoint upon arrival. To capture the second intruder, $\mathcal{O}$ moves the vehicle for $1+\rho$ time units, arriving at the perimeter at exactly time $2+\rho$. The vehicle can capture the second intruder given the definition of $\epsilon$ yielding $\mathbb{E}_{\bar{X}}[n_{\mathcal{O}}] = 2$.

We now describe the performance, in expectation, of a deterministic algorithm $\mathcal{A}$ against the same probability distribution $\bar{X}$ on $\mathcal{I}$. Observe that the best way to capture both intruders is to move the vehicle to one of the endpoint. However, as $\mathcal{A}$ operates online, $\mathcal{A}$

does not know the arrival time and location of the intruders. Thus, given that the input sequence is random, the probability that $\mathcal{A}$ will have the vehicle at the correct endpoint at time 1 is 1/2. More precisely, assume that $\mathcal{A}$ moves the vehicle to endpoint +1 by time +1. Then, against input $\mathcal{I}_1$, the vehicle can capture both $i^1_{1,+1}$ and $i^{1+\epsilon}_{1,-1}$ by moving towards $-\rho$ after capturing $i^1_{1,+1}$. However, against input $\mathcal{I}_2$, the vehicle can capture only $i^t_{1,-1}$ if $v = \frac{1-\rho}{1+\rho}$ holds. Otherwise, the vehicle captures only $i^{1+\epsilon}_{1,+1}$. Thus, $\mathbb{E}_{\bar{X}}[n_{\mathcal{A}}] = 1 + \frac{1}{2} = \frac{3}{2}$. Thus, from Lemma 21 we obtain the result. □

### 3.3 Randomized Algorithms

In this section, we design and analyze three randomized algorithms. Our first algorithm is an open loop and memoryless algorithm and highlights the advantage of using randomized algorithms. Specifically, we establish that this algorithm is 2-competitive for all values of $v$ and $\rho$.

#### 3.3.1 Algorithm Stay

Algorithm Stay is summarized as follows. At time 0, assuming that the vehicle is located at the origin, the vehicle randomly selects a location out of the two choices $\{-\rho, \rho\}$, each having equal probability of 0.5. Once a location is picked, the vehicle moves to that location and stays at that location until the arrival of the intruders end.

**Theorem 26.** *Algorithm Stay is* 2*-competitive against an oblivious adversary for any problem instance* $\mathcal{P}$*.*

*Proof.* Let $R \geq 0$ and $L \geq 0$ denote the total number of intruders preplanned to be released at endpoints +1 and −1, respectively, in the time interval $[0, T]$, where $T > 0$ denotes the time instant until which the intruders are released. Then, the expected number of intruders captured by the vehicle is $0.5\frac{R}{R+L} + 0.5\frac{L}{R+L} = 0.5$. Thus, from Definition 1 and by assuming that the optimal offline algorithm can capture all, we obtain the claim that Algorithm Stay is 2-competitive.

We now provide an input sequence $\mathcal{I}$ to establish that the bound is tight. Input sequence $\mathcal{I}$ begins at time $\rho$ with $0 < R \leq 2N$ intruders arriving at endpoint $+1$ and no intruders arriving at endpoint $-1$. The number of intruders captured by Algorithm Stay, in expectation, is $\frac{R}{2}$. On the other hand, since an optimal offline algorithm has prior information about the input sequence, it captures all $R$ intruders by moving the vehicle to $\rho$. Thus, from Definition 3, the competitive ratio is exactly 2 and hence the bound obtained is tight. □

Recall that in Chapter 1, we introduced an analogous algorithm called Simple Algorithm in Example 1 with the key difference that Simple Algorithm was deterministic. By selecting a location randomly instead of deterministically, Algorithm Stay achieves a competitive ratio of 2 whereas Simple Algorithm was not $c$-competitive. This highlights the effect of using randomization to design online algorithms. We now describe our second algorithm that randomly selects the initial location of the vehicle from a set of locations.

### 3.3.2 Randomized Stay Algorithm

At time 0 or at the start of the first epoch, the vehicle is either located at endpoint $+1$ or $-1$. The choice of the endpoint is random, i.e., with probability 0.5, the vehicle is located at endpoint $+1$ and with probability 0.5, the vehicle is located at $-1$. The vehicle then waits until the first intruder arrives in the environment. Denote the time instant when the first intruder arrives as $t \geq 0$. At time instant $t$, with probability 0.5, the vehicle either stays at the endpoint until time $t+2$ or, with probability 0.5, moves to the other endpoint. Note that the vehicle reaches the other endpoint by exactly time $t+2$. At time $t+2$, the vehicle then repeats, i.e., the first epoch ends and the second epoch begins.

**Lemma 27.** *Suppose that the vehicle is located at endpoint $+1$ (resp. $-1$) at time instant $t$ and $l_t$ (resp. $r_t$) intruders arrive at endpoint $-1$ (resp. $+1$) at time $t$. Then the vehicle can capture $l_t$ (resp. $r_t$) intruder if*

$$v \leq \frac{1-\rho}{1+\rho}.$$

*Proof.* Suppose that the vehicle is located at endpoint $+1$ and $l_t$ intruders arrive at $-1$ at time $t$. These intruders will reach the perimeter $-\rho$ at time $t + \frac{1-\rho}{v}$. The vehicle can reach the perimeter $-\rho$ at time $t+1+\rho$ as it first moves the origin and then to location $-\rho$. Thus, the vehicle can capture the intruder released at time $t$ if $1+\rho \leq \frac{1-\rho}{v}$. $\square$

**Theorem 28.** *For any problem instance $\mathcal{P}$ for which $v \leq \frac{1-\rho}{1+\rho}$ holds,*

$$c_{Rand-Stay} = \begin{cases} \frac{1+(\lfloor\frac{2}{\Delta}\rfloor-1)2N}{\frac{3}{4}+(\lfloor\frac{2}{\Delta}\rfloor-1)N}, & \text{if } \frac{2}{\Delta} \in \mathbb{Z}_+ \\ \frac{1+\lfloor\frac{2}{\Delta}\rfloor 2N}{\frac{3}{4}+\lfloor\frac{2}{\Delta}\rfloor N}, & \text{otherwise.} \end{cases}$$

*Proof.* Suppose that the vehicle selects endpoint $+1$ at time instant 0 and $r_t$ (resp. $l_t$) intruders arrive at endpoint $+1$ (resp. $-1$) at time $t \geq 0$. First consider that the vehicle decides to move towards endpoint $-1$ at time instant $t$. As the vehicle waits until the first intruder is released, at least one out of $r_t$ and $l_t$ must be strictly greater than 0. Further, the vehicle captures $r_t$ intruders upon arrival. Let $p = \lfloor\frac{2}{\Delta}\rfloor - 1$ if $\frac{2}{\Delta}$ is an integer and $p = \lfloor\frac{2}{\Delta}\rfloor$, otherwise. Denote $r_1, r_2, \dots, r_p$ (resp. $l_1, l_2, \dots, l_p$) as the intruders that are released at endpoint $+1$ (resp. $-1$) during the time interval $[t+\Delta, t+2)$. From Lemma 27, the vehicle can capture all the intruders that are released at endpoint $-1$ in the time interval $[t+\Delta, t+2)$. Thus, in this case, the vehicle captures $\frac{1}{4}[r_t + l_t + l_1 + \cdots + l_p]$, where $\frac{1}{4}$ is the probability that the vehicle selects endpoint $+1$ and decides to move towards endpoint $-1$. We now consider that the vehicle decides to stay at endpoint $+1$ until time $t+2$. In this case, the vehicle captures $\frac{1}{4}[r_t + r_1 + \cdots + r_p]$. The number of intruders captured when the vehicle selects endpoint $-1$ at time 0 can be determined analogously. Since the vehicle selects an endpoint with probability 0.5 and then decides to stay or move to the other endpoint with probability 0.5, it follows that the expected number of intruders that the vehicle captures in the time interval $[t, t+2)$ is

$$\frac{3}{4}(r_t + l_t) + \frac{1}{2}\Big[\sum_{q=1}^{p} r_q + l_q\Big].$$

**Algorithm 4:** Sleep-Switch Algorithm

1. Select endpoint $+1$ or $-1$ with probability 0.5.
2. Select sleep time $s$ from $\{0, \Delta, \ldots, p\Delta\}$ with probability $\frac{1}{p+1}$.
3. Wait until the first intruder arrives.
4. Reset time to 0 and sleep until time $s$ at endpoint.
5. Move to other endpoint.
6. Repeat from Step 3.

At time $t+2$, the probability of the vehicle to be at endpoint $+1$ and $-1$ is 0.5 each. Thus, following analogous steps as before, it can be shown that for any epoch $k \geq 1$, the expected number of intruders captured by the vehicle is $\frac{3}{4}(r_t + l_t) + \frac{1}{2}[\sum_{q=1}^{p} r_q + l_q]$. Finally, assuming that there exists an optimal offline algorithm that can capture all intruders yields

$$c_{Rand-Stay} = \frac{r_t + l_t + \sum_{q=1}^{p} r_q + l_q}{\frac{3}{4}(r_t + l_t) + \frac{1}{2}\Big[\sum_{q=1}^{p} r_q + l_q\Big]} \leq \frac{1 + 2pN}{\frac{3}{4} + pN}.$$

This concludes the proof. □

Observe that Algorithm Rand-Stay achieves the same performance as Algorithm Stay if $pN$ tends to infinity and achieves the fundamental limit if $pN$ tends to zero.

We now present an algorithm that has a competitive ratio strictly lower than that of Algorithm Rand-Stay but requires the information of parameter $\Delta$.

### 3.3.3 Algorithm Sleep-Switch

Contrary to the previous algorithms, Algorithm Sleep-Switch requires the information of the parameter $\Delta$. Algorithm Sleep-Switch is defined in Algorithm 4 and summarized as follows.

Algorithm Sleep-Switch first selects an endpoint randomly with equal probability of 0.5. Then, with the same $p$ as defined for in Algorithm Rand-Stay, Algorithm Sleep-Switch selects a sleep time $s$ from a set of sleep times $S := \{0, \Delta, 2\Delta, \ldots, p\Delta\}$ with equal probability of $\frac{1}{p+1}$ and stays at the selected endpoint until time $s$. Suppose that the vehicle is located at endpoint $+1$. Then, at time $s$, the vehicle moves to endpoint $-1$. The vehicle then sleeps again for the same amount $s$ of time units and repeats the process.

**Theorem 29.** *For any problem instance $\mathcal{P}$, Algorithm Sleep-Switch is $\frac{2p+2}{p+2}$-competitive if $v \leq \frac{1-\rho}{1+\rho}$ holds.*

*Proof.* Suppose that the vehicle is at endpoint $+1$ at time $k_s$, where $k_s$ denotes the time when an epoch $k$ begins. Let $r_0, \ldots, r_p$ (resp. $l_0, \ldots, l_p$) denote the intruders that arrive at $+1$ (resp. $-1$) at time instants $0, \Delta, \ldots, p\Delta$, respectively. Similarly, let $r_{p+1}, r_{p+2}, \ldots, r_{2p+1}$ (resp. $l_{p+1}, l_{p+2}, \ldots, l_{2p+1}$) denote the intruders that arrive at time $(p+1)\Delta, (p+2)\Delta, \ldots, (2p+1)\Delta$, respectively. Note that $(p+1)\Delta \geq 2$. For any sleep time $s \in \{0, \ldots, p\Delta\}$, the expected number of intruders captured by the vehicle in time interval $[k_s, k_s + s + 2)$ is

$$\frac{1}{2}\left(\sum_{q=0}^{p} r_q + l_q + l_s + r_s + \sum_{q=p+1}^{p+s} l_q + r_q\right),$$

where we used the fact that the probability of vehicle being at endpoint $+1$ and endpoint $-1$ is 0.5. Observe that the $\sum_{q=p+s+1}^{2p+1} l_q + r_q$ intruders are not lost by time $k_s + s + 2$ and can be captured in the next epoch. Thus, the number of intruders captured by the vehicle in time interval $[k_s, k_s + s + 2)$ is

$$\frac{1}{2}\left(\sum_{q=0}^{p} r_q + l_q + l_s + r_s + \sum_{q=p+1}^{p+s} l_q + r_q + \sum_{q=p+s+1}^{2p+1} l_q + r_q\right) = \frac{1}{2}\left(\sum_{q=0}^{2p+1} r_q + l_q + r_s + l_s\right),$$

where $\sum_{q=p+s+1}^{2p+1} l_q + r_q$ denotes the number of intruders captured by the vehicle in the next epoch. Since the sleep time $s$ is chosen with probability $\frac{1}{p+1}$, the expected number of intruders captured in an epoch is

$$\frac{(p+2)(\sum_{q=0}^{2p+1} r_q + l_q)}{2(p+1)}.$$

Finally, assuming that there exists an optimal offline algorithm that can capture all intruders yields the result. □

We now provide a parameter regime plot highlighting the parameter regime in which our algorithms are effective as well as the parameter regime for the established necessary conditions.

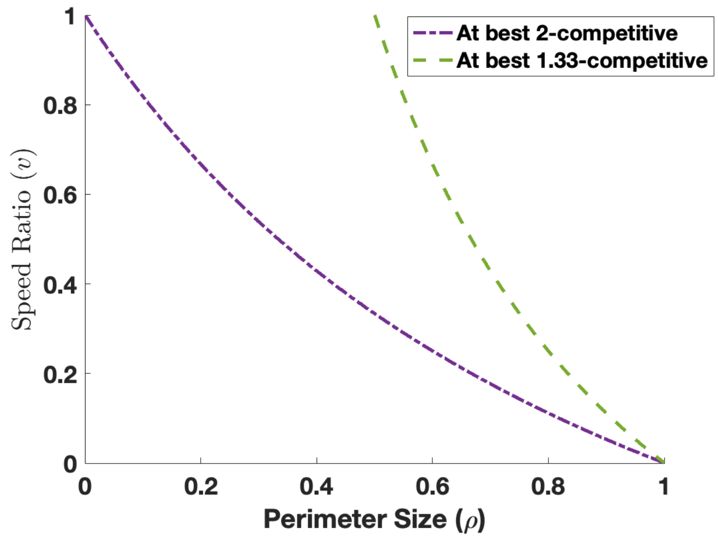



Figure 3.1 Parameter regime plot highlighting the fundamental limits.

### 3.4 Results and Discussions

Figure 3.1 shows the $(v, \rho)$ plot summarizing our results. The area above the green dashed curve denotes the fundamental limit established in Theorem 24. Similarly, the area above the purple dashed curve denotes the fundamental limit established in Theorem 25 as well as Theorem 23. Although the parameter space for Theorem 25 and Theorem 23 is the same, the lower bound on the achievable competitive ratio is different.

The area below the purple dashed curve denotes the effective parameter space for Randomized Stay and Sleep Switch algorithms. Specifically, for any values of perimeter size $\rho$ and speed ratio $v$ that is below the purple curve, Algorithm Randomized Stay and Algorithm Sleep Switch have a finite competitive ratio as established in Theorem 28 and Theorem 29, respectively. Note that algorithm Stay is 2-competitive in the entire parameter space, i.e., all values of $v$ and $\rho$ in Figure 3.1.

### 3.5 Conclusion

We study a problem where a single defender defends a linear perimeter from mobile intruders that move inwards towards the perimeter upon arrival. A key contribution is the combination of concepts and techniques from competitive analysis of randomized online algorithms with pursuit of multiple mobile intruders. We designed and analyzed three randomized algorithms for incoming intruders. The first algorithm is demonstrated to be 2-competitive in the entire parameter space whereas the competitive ratio of the other two algorithms vary with the problem parameters. We also derived a fundamental limits on the existence of randomized algorithms having competitive ratio of at least 2 and another fundamental limit on the existence of randomized algorithms having competitive ratio of at least 1.33. Finally, we derived a fundamental limit on the existence of zealous algorithms having competitive ratio of at least 2, when arbitrary many intruders can be released in the environment.

# CHAPTER 4

# COMPETITIVE PERIMETER DEFENSE IN CONICAL ENVIRONMENTS

In this chapter, we extend the perimeter defense problem considered in Chapter 2 to planar environments and multiple defenders. The environment contains $M \geq 1$ defenders which seek to defend a perimeter by intercepting mobile intruders (cf. Fig. 4.1). The intruders are released at the boundary of the environment and move radially inwards with fixed speed toward the perimeter. The defenders, each having a finite capture radius and having access to intruder locations only after they are released, move with unit speed and with the aim of capturing as many intruders as possible before they reach the perimeter. This is an *online* problem as the input, i.e., the release times, numbers of intruders and their release locations, is gradually revealed over time. Thus, we focus on the design and analysis of online algorithms to route the defenders. Aside from military applications, this scenario models relaying critical information to targets and defending wildlife against poachers [22].

The key question that is addressed in this chapter is how does the analytical bound on the competitiveness scale with multiple defenders? The main contributions of this work are as follows:

1. **Perimeter defense problem with $M$ defenders:** We address a perimeter defense problem in a conical environment with $M \geq 1$ identical defenders tasked to defend a perimeter. Each defender has a finite capture radius and moves with unit speed. We do not impose any assumption on the arrival process of the intruders. More precisely, an arbitrary number of intruders can be released in the environment at arbitrary locations and time instances. Upon release, the intruders move with fixed speed $v$ towards the perimeter. Thus, the perimeter defense problem is characterized by five parameters: (i) angle $\theta$ of the conical environment, (ii) the speed ratio $v$ between the intruders and the defenders, (iii) the perimeter radius $\rho$, (iv) the number of defenders $M$, and (v) the capture radius of the defenders $r$.

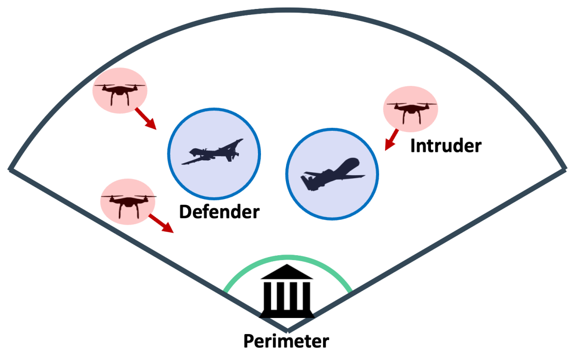



Figure 4.1 Illustration of the problem considered in this work. Multiple defenders (light blue disc) seek to defend a perimeter (green arc) from arbitrary many intruders (light red disc) that arrive over time in the conical environment. The capture radius of the defenders is shown by the dark blue circle surrounding the defenders.

2. **Necessary condition:** We establish a necessary condition on competitiveness of any online algorithm. Specifically, we establish a necessary condition on the existence of any $c$-competitive algorithm for any finite $c$. We also characterize the minimum number of defenders required to capture all intruders.

3. **Algorithm Design and Analysis:** We first design and analyze three classes of decentralized algorithms. We then design and analyze two classes of cooperative algorithms. Specifically, the first two decentralized algorithms are provably 1 and 2-competitive, respectively and the third decentralized algorithm exhibits a finite competitive ratio which depends on the problem parameters. Similarly, the first cooperative algorithm is provably 1.5-competitive and the second cooperative algorithm exhibits a finite competitive ratio which depends on the problem parameters.

Additionally, through multiple parameter regime plots, we shed light into the relative comparison and the effectiveness of our algorithms. We also provide an insight into how the results in this work can be extended when the defenders have different capture radii.

This chapter, content of which is based on [10, 6], is organized as follows. In Section 4.1, we formally define the problem. Section 4.2 establishes the necessary conditions, Section 4.3 presents the algorithms and their analysis. Section 4.4 provides several numerical insights and

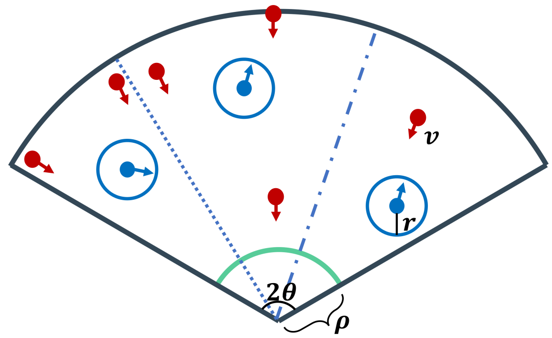



Figure 4.2 Problem description for $M = 3$ defenders. The red dots denote the intruders whereas the blue dots and blue circles denote the defenders and the capture circle, respectively. The perimeter is depicted by the green curve. The first ($m = 1$) partition contains two intruders, second ($m = 2$) partition contains three intruders and the third ($m = 3$) partition contains one intruder. The common boundary of $V_1$ and $V_2$ is denoted as the blue dot line and the common boundary of $V_2$ and $V_3$ is denoted as the blue dashed-dot line, respectively.

outlines an extension to the case of different capture radii. Finally, Section 4.5 summarizes the chapter.

## 4.1 Problem Formulation

Consider a conical environment of $\mathcal{E}(\theta) = \{(y, \alpha) : 0 < y \leq 1, -\theta \leq \alpha \leq \theta\}$ which contains a conical region $\mathcal{R}(\rho, \theta) = \{(z, \alpha) : 0 < z \leq \rho < 1, -\theta \leq \alpha \leq \theta\}$, where $\theta$ is measured with respect to $y-$axis. An arbitrary number of intruders are released at arbitrary time instants at the circumference of the environment, i.e., $y = 1$. Each intruder moves radially with a fixed speed $v < 1$ towards the origin in order to breach the *perimeter*. The perimeter is defined as the boundary of $\mathcal{R}(\rho, \theta)$, i.e., $\partial\mathcal{R}(\rho, \theta) = \{(y, \alpha) : y = \rho, -\theta \leq \alpha \leq \theta\}$ (cf Fig. 4.2). We consider $M \geq 1$ identical defenders having their motion modeled as a first order integrator[1], tasked to defend the perimeter. Each defender can move with unit speed[2] and has a finite capture radius $r < \rho$ as the problem is trivial for any $r \geq \rho$. Later, in Lemma 32, we will characterize even lower values of $r$ for which problem is trivial. A *capture circle* of a defender is defined as a circle of radius $r$, centered at the defender's location. An intruder is *captured* and subsequently removed from $\mathcal{E}(\theta)$ if it lies within or on the capture circle. An

[1]This work can be extended to other models such as double integrator and will be considered in the future.

[2]The speed is set to unity to normalize the speed of the intruders. Thus we use the two terms, speed of the intruders and speed ratio, interchangeably.

intruder is *lost* if it reaches the perimeter without being captured by the defender. Thus, a *problem instance* $\mathcal{P}(\theta, \rho, v, r, M)$ for this problem is characterized by five parameters: the speed of the intruders (or speed ratio), $v < 1$, the perimeter's radius $0 < \rho < 1$, the angle that defines the size of the environment as well as the perimeter, $0 < \theta \leq \pi$, the capture radius of each of the defender $r < \rho$, and the number of defenders $M > 1$. For compactness, we will denote a problem instance as $\mathcal{P}$.

**Problem Statement:** The aim is to understand how the number of defenders effects the competitive ratio and the parameter regimes. Specifically, the key objectives of this work are the following:

1. Establish a necessary condition for the existence of online algorithms with a finite competitive ratio.
2. Design and analyze algorithms with provable guarantees on the competitive ratio.
3. Characterize the dependence of the competitive ratio on parameter regimes, especially $M$.

In the next section, we establish a necessary condition on the existence of a $c$-competitive algorithm for any finite $c \geq 1$.

### 4.2 Fundamental Limits

As our analysis relies heavily upon partitioning the environment, we start by formally defining a *partition* of the environment.

**Definition 30** (Partition of $\mathcal{E}(\theta)$)**.** *A partition of $\mathcal{E}(\theta)$ is a collection of $q \geq 1$ cones $\mathcal{W} = \{W_1, W_2, \ldots, W_q\}$ of unit radius with vertex at the origin, disjoint interiors, and whose union is $\mathcal{E}(\theta)$. We refer to a cone $W_m, 1 \leq m \leq q$ as the m-th dominance region.*

We first observe that for any set of initial locations of the $M$ vehicles in which no two vehicles have the same angular coordinate, there exists a partition of the environment with $M$ dominance regions such that each dominance region can be assigned to a particular vehicle.

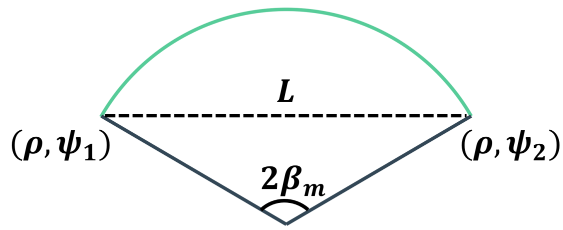



Figure 4.3 Blowout image of the perimeter contained in the $m$-th partition for the description of proof of Lemma 31 for $m$-th partition.

Given $M$ dominance regions $\{W_1, \ldots, W_M\}$, with each dominance region corresponding to a vehicle, we refer to an individual vehicle $V_m, 1 \leq m \leq M$ based on the dominance region it occupies, unless stated otherwise. More precisely, we denote a vehicle as $V_m$, $1 \leq m \leq M$, associated with the $m$-th dominance region. We assume that the dominance regions are numbered from left to right when $\theta < \pi$ and in clockwise order if $\theta = \pi$ (cf. Fig. 4.2). Further, two vehicles $V_m$ and $V_{m+1}$ for any $1 \leq m \leq M-1$ are called *neighbors if their dominance regions share a boundary.* A *common boundary* for neighbors $V_m$ and $V_{m+1}$ is the rightmost (resp. leftmost) boundary of the $m$-th (resp. $m+1$-th) dominance region (cf. Fig. 4.2). Finally, we denote the portion of the perimeter $\partial R(\rho, \theta)$ corresponding to dominance region $m, 1 \leq m \leq M$ as $\partial R_m$.

We now characterize the locations for each vehicle $V_m, 1 \leq m \leq M$ such that the distance from that location to any location on $\partial R_m$ is the least.

**Lemma 31.** *For a problem instance $\mathcal{P}$, positioning each defender $V_m$, $1 \leq m \leq M$, at locations $(\rho \cos(\frac{\theta}{M}), -(M-2m+1)\frac{\theta}{M})$ minimizes the distance, and equivalently the time taken, from the defender's location to any location on $\partial R_m$.*

*Proof.* Let $(x_m, \alpha_m) \in \mathcal{E}(\theta)$ denote the location for vehicle $V_m, \forall m \in \{1, \ldots, M-1\}$, such that the distance from location $(x_m, \alpha_m)$ to any location on $\partial R_m$ is the least.

Consider the $m$-th dominance region of angle $2\beta_m$ that contains the vehicle $V_m$ and let $L$ be the line joining the two points $(\rho, \psi_1)$ and $(\rho, \psi_2)$ such that $\psi_2 - \psi_1 = 2\beta_m$ and $\psi_2 > \psi_1$, which correspond to the two endpoints of $\partial R_m$ (cf. Fig. 4.3). To minimize the distance to any location on $\partial R_m$ from the location $(x_m, \alpha_m)$, the vehicle $V_m$ must be located at the midpoint of line $L$, i.e., at the angle bisector of angle $2\beta_m$. Further, since at least one out of the two locations $(\rho, \psi_1)$ and $(\rho, \psi_2)$ is shared by the neighbors of $V_m$, we require that the distance to the location $(\rho, \psi_1)$ (resp. $(\rho, \psi_2)$) for both $V_m$ and $V_{m-1}$ (resp. $V_{m+1}$) vehicles must be equal. Without loss of generality, we assume that $(\rho, \psi_2)$ is shared by vehicle $V_m$ and $V_{m+1}$. Then,

$$\rho \sin(\beta_m) - r = \rho \sin(\beta_{m+1}) - r \Rightarrow \beta_m = \beta_{m+1}, \tag{4.1}$$

where $2\beta_{m+1}$ is the angle of the $(m+1)$-th dominance region. Equation (4.1) must hold for each $1 \leq m \leq M-1$, which yields $\beta_1 = \beta_2 = \cdots = \beta_M$. Observing that $\sum_{i=1}^{M} 2\beta_i = 2\theta$ we obtain $\beta_m = \frac{\theta}{M}$ for every $1 \leq m \leq M$. In other words, this means that every dominance region $m \in \{1, \ldots, M\}$ must have an angle of $\frac{2\theta}{M}$. Combining this fact with the fact that each vehicle $V_m$ must be located at the angle bisector of the $m$-th dominance region, it follows that $\alpha_m = -(M - 2m + 1)\frac{\theta}{M}$. Finally, using basic trigonometric identities, we obtain $x_m = \rho \cos(\frac{\theta}{M})$. □

Lemma 3 formally established the intuitive conclusion that $\mathcal{E}$ must be equally partitioned between the vehicles. The next result characterizes a value for the capture radius for which all intruders released in the environment can be captured by simply keeping the vehicles stationary at specific locations.

Our next result characterizes the minimum capture radius required, given $M$ defenders, such that all intruders that arrive in the environment are captured. Equivalently, the result characterizes the minimum number of defenders required, given the capture radius of each defender, such that all intruders that arrive in the environment are captured.

**Lemma 32.** *For any problem instance $\mathcal{P}$ with $r \geq \rho \sin(\frac{\theta}{M})$, all intruders can be captured by keeping vehicle $V_m$ stationary at location $\left(\rho \cos(\frac{\theta}{M}), -(M - 2m + 1)\frac{\theta}{M}\right)$, $\forall 1 \leq m \leq M$.*

*Proof.* Recall from Lemma 31 that the perimeter can be partitioned into $M$ partitions, each of angle $\frac{2\theta}{M}$. The idea is to determine the location $(x_m, \alpha_m) \in \mathcal{E}(\theta)$ for defender $V_m$ $\forall m \in \{1, \ldots, M\}$ such that when $V_m$ is located at $(x_m, \alpha_m)$, $\partial R_m$ is completely contained within

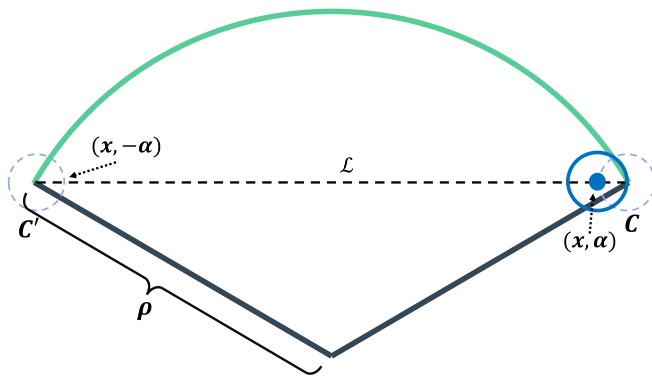



Figure 4.4 Description of proof of Lemma 33. The blue dashed circles $\mathcal{C}$ and $\mathcal{C}'$ are centered at $(\rho, \theta)$ and $(\rho, -\theta)$ respectively. The defender, denoted by the blue dot, is located at $(x, \alpha)$. The line $\mathcal{L}$ is denoted by the black dashed line.

the capture circle of $V_m$. To ensure this, the two end points of $\partial R_m$ must be equidistant from $(x_m, \alpha_m)$, which implies that the defender $V_m$ must be located at the angle bisector of the dominance region. This yields $\alpha_m = -(M - 2m + 1)\frac{\theta}{M}$. Finally, applying trigonometric identities yields $r = \rho \sin(\frac{\theta}{M})$. This concludes the proof. □

In light of Lemma 32, we assume that $r < \rho \sin(\theta/M)$ in the remainder of this chapter. Our next result will be useful in establishing the necessary condition for the special case, $M = 1$.

**Lemma 33.** *Suppose that $M = 1$. Then the minimum time required by the defender to move from a location such that the capture circle contains one end of the perimeter, $(\rho, \theta)$, to a location such that the capture circle contains the opposite end of the perimeter, $(\rho, -\theta)$, is $2(\rho \sin(\theta) - r)$ if $\theta < \frac{\pi}{2}$ and $2(\rho - r)$, otherwise.*

*Proof.* Consider two circles $\mathcal{C}$ and $\mathcal{C}'$, each of radius $r$ and with centers coinciding with the points $(\rho, \theta)$ and $(\rho, -\theta)$, respectively (Fig. 4.4). Observe that, if the defender is located at any point within the intersection of the circle $\mathcal{C}$ (resp. $\mathcal{C}'$) and the perimeter, then the location $(\rho, \theta)$ (resp. $(\rho, -\theta)$) will be contained within the capture circle.

Let $(x, \alpha) \in \mathcal{E}(\theta)$ (resp. $(x, -\alpha)$) denote a point on the circumference of circle $\mathcal{C}$ (resp. $\mathcal{C}'$). Our aim is to determine the closest possible pair of locations $(x, \alpha)$ and $(x, -\alpha)$; that is, the pair of positions that minimizes the distance the defender needs to travel to go from one point to the other. We consider two cases: (i) $\theta \leq \frac{\pi}{2}$ and (ii) $\theta > \frac{\pi}{2}$.

For case (i), finding the closest pair of points corresponds to determining the shortest distance along the line $\mathcal{L}$ between the two circles $\mathcal{C}$ and $\mathcal{C}'$. The shortest line that connects any two non intersecting circles must pass through the center of the circles, so $\mathcal{L}$ must pass through $(\rho, \theta)$ and $(\rho, -\theta)$, and the points $(x, \alpha)$ and $(x, -\alpha)$ are where $\mathcal{L}$ intersects $\mathcal{C}$ and $\mathcal{C}'$, respectively. We compute the distance between the two points as follows. We first find the angle bisector which bisects the angle $2\theta$ of the environment. Note that the angle bisector is also the perpendicular bisector of line $\mathcal{L}$ as the triangle formed by joining points $(\rho, \theta)$, $(\rho, -\theta)$ and the origin is an isosceles triangle. By constructing a triangle that joins the points $(x, \alpha)$, origin, and the midpoint of line $\mathcal{L}$ and using trigonometric identities, we determine that $x = \frac{\rho \sin(\theta) - r}{\sin(\alpha)}$ and $\alpha = \tan^{-1}(\frac{\rho \sin(\theta) - r}{\rho \cos(\theta)})$. From geometry, it follows that the length of the line segment joining the points $(x, \alpha)$ and $(x, -\alpha)$ is $2(\rho \sin(\theta) - r)$. Note that when $\theta = \frac{\pi}{2}$, $\mathcal{L}$ runs through the origin simplifying this expression to $2(\rho - r)$ matching the expression from case (ii).

For case (ii), the corresponding line $\mathcal{L}$ is not contained in the environment which means the defender cannot travel on $\mathcal{L}$ to move from $(x, \alpha)$ to $(x, -\alpha)$. Instead, the shortest path is to first move to the origin from $(x, \alpha)$ and then to the location $(x, -\alpha)$ from the origin. This gives us $x = \rho - r$ and $\alpha = \theta$ and a minimum distance of $2(\rho - r)$. This concludes the proof. □

We now establish a necessary condition for the existence of $c$-competitive algorithms. We first establish the result for $M = 1$ and the for any $M > 1$.

**Theorem 34.** ***(Necessary condition for finite*** $c(\mathcal{P})$***)*** *For any problem instance* $\mathcal{P}$ *with* $M = 1$ *and parameters satisfying*

$$2(\rho \sin(\theta) - r) > \frac{1 - \rho}{v},\ \textit{if}\ \theta < \frac{\pi}{2},$$
$$2(\rho - r) > \frac{1 - \rho}{v},\ \textit{if}\ \theta \geq \frac{\pi}{2},$$

*there does not exist a c-competitive algorithm for any constant c and no algorithm, either online or offline, can capture all intruders.*

*Proof.* Recall from Definition 1 that any online algorithm $\mathcal{A}$ is $c$-competitive if the condition $n_{\mathcal{O}}(\mathcal{I},\mathcal{P}) \leq cn_{\mathcal{A}}(\mathcal{I},\mathcal{P})$ holds for *every* input sequence $\mathcal{I}$. Thus, the aim is to construct an input sequence $\mathcal{I}$ such that the condition $n_{\mathcal{O}}(\mathcal{I},\mathcal{P}) \leq cn_{\mathcal{A}}(\mathcal{I},\mathcal{P})$ does not hold for any constant $c \geq 1$ and for every online algorithm.

For both $\mathcal{A}$ and optimal offline $\mathcal{O}$ algorithms, assume that the defender starts at the origin at time 0. The input sequence starts at time instant 1 with a *stream* of intruders, i.e., a single intruder being released every $\frac{1-\rho}{v}$ time units apart, at location $(1,\theta)$. If $\mathcal{A}$ never captures any stream intruders, the stream never ends meaning the algorithm $\mathcal{A}$ will not be $c$-competitive for any constant $c \geq 1$, and the first result follows as $\mathcal{O}$ can move to $(\rho,\theta)$ and capture all the stream intruders. We thus assume $\mathcal{A}$ does capture at least one stream intruder, say the $i^{\text{th}}$ one, at time $t$. The input instance ends with the release of a burst of $c+1$ intruders that arrive at location $(1,-\theta)$ at the same time instant $t$.

We now identify how many intruders $\mathcal{A}$ can capture. First, it cannot capture stream intruders 1 through $i-1$ because the stream intruders arrive $\frac{1-\rho}{v}$ time units apart meaning the previous intruder reaches the perimeter and thus is lost just as the next stream intruder arrives. We now show that the defender cannot capture any of the $c+1$ burst intruders. At time $t$, the defender must be at most $r$ distance away from the $i^{\text{th}}$ stream intruder in order to capture it. Likewise, it has only $\frac{1-\rho}{v}$ time to move to capture the $c+1$ burst intruders that arrived at time $t$. Given the conditions, $2(\rho\sin(\theta)-r) > \frac{1-\rho}{v}$ (resp. $2(\rho-r) > \frac{1-\rho}{v}$) for $\theta < \frac{\pi}{2}$ (resp. $\theta \geq \frac{\pi}{2}$), the defender is ensured to lose the burst intruders.

On the other hand, the optimal offline algorithm $\mathcal{O}$ can move the defender to location $(x,\alpha)$, as defined in Lemma 33, until the first $i-1$ intruders have been captured and then move the defender to $(x,-\alpha)$ capturing the burst intruders, losing only the $i^{th}$ intruder. This concludes the proof. □

**Theorem 35** (**Necessary condition for finite** $c(\mathcal{P})$)**.** *For any problem instance $\mathcal{P}$ with $M > 1$, let $\alpha = \alpha^*$ denote a solution to the equation*

$$2\sin(\theta - (M-1)\alpha) - \sin(\alpha) - \frac{r}{\rho} = 0.$$

*Further, suppose that*

$$v > \begin{cases} \frac{1-\rho}{2(\rho\sin(\theta-(M-1)\alpha^*)-r)}, & \text{if } 0 < \alpha^* < \frac{\pi}{2}, \\ \frac{1-\rho}{\rho-r}, & \text{otherwise.} \end{cases}$$

*Then, there does not exist a c-competitive algorithm, for any finite c.*

*Proof.* Recall from Definition 1 that any online algorithm $\mathcal{A}$ is $c$-competitive if the condition $n_{\mathcal{O}}(\mathcal{I}) \leq c n_{\mathcal{A}}(\mathcal{I})$ holds for *every* input sequence $\mathcal{I}$. Thus, the aim is to construct an input sequence $\mathcal{I}$ such that the condition $n_{\mathcal{O}}(\mathcal{I}) \leq c n_{\mathcal{A}}(\mathcal{I})$ does not hold for any constant $c \geq 1$ and for every online algorithm. The proof is divided into three parts. First, we describe a family of $M$ input sequences. We will show that any one input sequence from these $M$ inputs yields the same result. Then, we characterize the best locations for the vehicles against these input sequence. Finally, we determine the performance of the best online algorithm as well as an optimal offline algorithm to establish the result. We start by constructing the input sequences.

Let $\mathbf{I} = \{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_M\}$ denote a set or family of input sequences. Each input sequence $\mathcal{I}_l \in \mathbf{I}$, where $l \in \{1, \dots, M\}$, differs in the release location of intruders. Without loss of generality, we assume that both $\mathcal{A}$ and $\mathcal{O}$ have all of their vehicles at the origin at time instant 0.

Every input sequence $\mathcal{I}_l \in \mathbf{I}$ starts at time instant 1 and consists of a stream of intruders, i.e., a sequence of a single intruder arriving at location $(1, \theta)$ at every time instant $1 + k\frac{1-\rho}{v}, k \in \mathbb{N} \cup \{0\}$ until time instant $t$. The time instant $t \geq 0$ corresponds to the time instant when a vehicle $V_m$, for any $1 \leq m \leq M$, captures an intruder from the stream. A burst of $c+1$ intruders is then released at time instant $t$. The location where the burst of intruder is

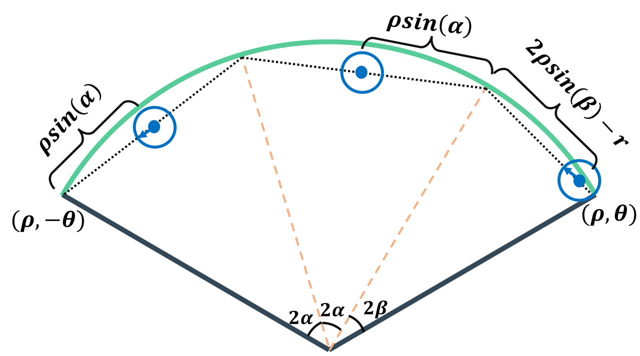



Figure 4.5 Snapshot of proof of Theorem 35 at time instant $t$. Vehicle $V_3$ has just captured $i$-th stream intruder. Note that $V_3$ has to move $2\rho\sin(\beta)-2r$ distance to capture intruders at $(\rho, \theta-2\beta)$.

released is different for each input sequence $\mathcal{I}_l$. Since there are $M$ dominance regions in the environment, there are exactly $M$ locations where the burst of intruders can arrive. These $M$ locations will have the same angular coordinate as the endpoints of each $\partial R_m, \forall 1 \leq m < M$, excluding $\theta$ and including $-\theta$. Without loss of generality, the burst of intruders are released at location $(1, -\theta)$ for $\mathcal{I}_1$.

Note that for any input sequence $\mathcal{I}_l$, if no vehicle captures the stream intruder, the stream never ends and the result follows as the optimal offline algorithm $\mathcal{O}$ can have one of its vehicles move to location $(\rho, \theta)$ starting at time 0 and thus, capture all stream intruders. Thus, without loss of generality we assume that vehicle $V_M$ captures the stream intruders at time instant $t$ according to any online algorithm $\mathcal{A}$. Note that since the stream intruders arrive every $\frac{1-\rho}{v}$ time units apart and stops when an intruder from the stream is captured, no online algorithm can capture more than 1 intruder from the stream. For some $i > 0$, let the $i$-th intruder be the stream intruder that was captured by $V_M$ at time $t$.

We first determine the best location for $V_M$ to capture the $i$-th stream intruder as well as the best location for the remaining $M-1$ vehicles at time instant $t$. Recall from Lemma 31 that for any $1 \leq m \leq M$, positioning vehicle $V_m$ at the midpoint of the line joining the two endpoints of $\partial R_m$ minimizes the distance to any location on $\partial R_m$. Thus, the best location for all $M$ vehicles at time instant $t$ is determined based on the following three observations (cf. Fig. 4.5):

- **O1:** Vehicle $V_M$ must be located on the line joining the two endpoints of the perimeter within its dominance region and at a distance $r$ from location $(\rho, \theta)$.
- **O2:** The time taken by $V_M$ to reach the other endpoint of the perimeter must be equal to the time taken by $V_{M-1}$ to reach the same location.
- **O3:** All other remaining $M-1$ vehicles must be located at the midpoints of the line joining the two endpoints of the perimeter contained in their dominance region.

Let $2\beta$ be the angle of $M$-th dominance region and, from observation **O3**, let $2\alpha$ be the angle of each of the $M-1$ dominance regions. From observations **O1** and **O2**, we obtain

$$2\rho\sin(\beta) - 2r = \rho\sin(\alpha) - r \Rightarrow \frac{r}{\rho} = 2\sin(\beta) - \sin(\alpha).$$

Using the fact that $2(M-1)\alpha + 2\beta = 2\theta$ yields

$$f(\alpha) := 2\sin(\theta - (M-1)\alpha) - \sin(\alpha) - \frac{r}{\rho} = 0. \tag{4.2}$$

Let $\alpha^*$ denote a solution of (4.2). We now establish that there exists an $\alpha^*$ that satisfies equation (4.2).

Notice that equation (4.2) is a continuous function of $\alpha$. Suppose that $\alpha = \epsilon$, where $\epsilon > 0$ is a very small number. Then, $f(\epsilon) = 2\sin(\theta - (M-1)\epsilon) - \frac{r}{\rho} > 0$ as $2\sin\theta > \frac{r}{\rho}$. Now consider that $\alpha = \frac{\theta}{M-1} - \epsilon$ for the same $\epsilon$. Then, $f(\frac{\theta}{M-1} - \epsilon) = 2\sin\epsilon - \sin(\frac{\theta}{(M-1)} - \epsilon) - \frac{r}{\rho} < 0$, for a sufficiently small $\epsilon$. This means that for a sufficiently small $\epsilon > 0$, $f(\cdot)$ changes its sign in the interval $[\epsilon, \frac{\theta}{M-1} - \epsilon]$. Thus, from Intermediate Value Theorem, it follows that there must exist an $\alpha^*$ such that $f(\alpha^*) = 0$. Further, as there may be more than one solution to equation (4.2), $\alpha^*$ corresponds to the smallest value among those multiple solutions.

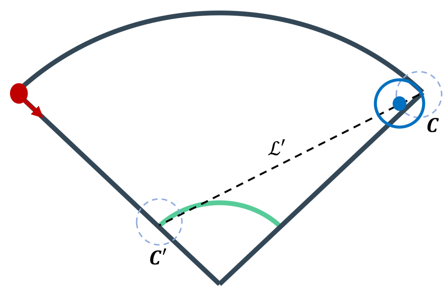
Let $\mathcal{A}_1$ denote any online algorithm that positions its vehicles as described above. We now describe the number of intruders captured by $\mathcal{A}_1$ on $\mathcal{I}_l \in \mathbf{I}$. Two cases arise based on the solution of equation (4.2); either $\alpha^* < \frac{\pi}{2}$ or $\alpha^* \geq \frac{\pi}{2}$. We first consider $\alpha^* < \frac{\pi}{2}$.

**Case 1** ($\alpha^* < \frac{\pi}{2}$): Observe that the solution of equation (4.2) yields angle $\alpha^*$ from which we compute the time taken, i.e., $2\rho\sin(\theta - (M-1)\alpha^*) - 2r$, by vehicle $V_M$ to reach the location $(\rho, \theta - 2\beta)$. Equivalently, from observations **O2** and **O3**, this expression also represents the time taken by any vehicle $V_m, 1 \leq m < M$, to reach an endpoint of $\partial R_m$. Thus, for any input sequence $\mathcal{I}_l \in \mathbf{I}$ releasing the $c+1$ intruders at time instant $t$ ensures that no vehicle can capture the burst intruders if $2\rho \sin(\theta - (M-1)\alpha^*) - 2r > \frac{1-\rho}{v}$. For $\mathcal{I}_1$, this location corresponds to $(1, -\theta)$.

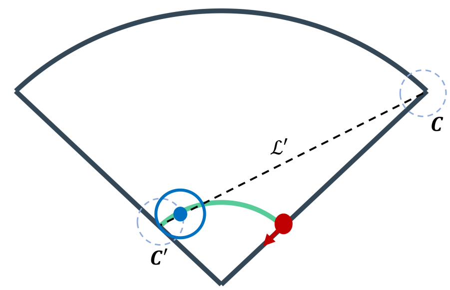
Before we consider Case 2, we briefly remark on online algorithms that positions $V_m, 1 \leq m < M$, vehicles at location other than the ones identified in Case 1, i.e., all online algorithms except $\mathcal{A}_1$. Recall that there are exactly $M$ locations for the burst to arrive in the environment. Of these $M$ locations, there exists at least one corresponding dominance region, say $\partial R_1$, having angle $2\alpha_1 > 2\alpha^*$. This implies that positioning the vehicle at the midpoint of the line joining the two endpoints $\partial R_1$, the vehicle will not be able to capture the burst intruders as $\rho \sin(\alpha_1) - r > \rho \sin(\alpha^*) - r$ holds. Thus, for any online algorithm that positions the $M$ vehicles at locations other than the location determined from the three observations, there exists an input sequence $\mathcal{I}_l \in \mathbf{I}$ for which the result still holds.

**Case 2** ($\alpha^* \geq \frac{\pi}{2}$): Given that $\alpha^* > 0$ and $M > 1$, this case is possible only when $M = 2$. This is because for any $M > 2$, $\beta > 0$, and $\alpha^* \geq \frac{\pi}{2}$, the condition $(M-1)\alpha^* + \beta = \theta$ does not hold since $\theta \leq \pi$. Thus for this case, we assume that $M = 2$. In this case, the line joining the two endpoints of the perimeter in the $(M-1)$-th dominance region either coincides with that of $M$-th dominance region (when $\theta = \pi$) or lies outside the environment (when $\theta < \pi$). Thus, it follows that the best position for $V_{M-1}$ is to be at the origin. Given the condition that $\rho - r > \frac{1-\rho}{v}$, vehicle $V_{M-1}$ cannot capture the $c+1$ intruders.

We now show that the optimal offline algorithm $\mathcal{O}$ captures all of the intruders on the input sequence $\mathcal{I}_l$. Recall that $\mathcal{O}$ has complete information of when, where, and how many intruders will arrive at time instant 0. Thus, $\mathcal{O}$ moves one of its vehicles, say $V_M$, to location $(\rho, \theta)$ to capture all $i$ intruders and at the same time $\mathcal{O}$ moves any one of the remaining $M-1$ vehicles, say $V_1$ to location $(\rho, \tau_l)$ capturing the $c+1$ intruders, where $\tau_l$ denotes the angular coordinate where the burst intruders is released in input sequence $\mathcal{I}_l$. Thus,




(a) The defender is located at $(t_1, \alpha_1)$ at time $t_1$. Intruder $b$ is at $(1, -\theta)$.




(b) The defender is located at $(t_2, \alpha_2)$. Intruder b is captured but intruder a is lost.

Figure 4.6 Description of the proof of Theorem 36 for $I_3$. The green curve denotes the perimeter. The circles $\mathcal{C}$ and $\mathcal{C}'$ are denoted by blue dashed circles and the line segment $\mathcal{L}'$ is denoted by the black dashed line.

$n_{\mathcal{O}}(\mathcal{I}) = i + c + 1$ and $n_{\mathcal{A}}(\mathcal{I}) = 1$ and the result follows. □

We highlight that the input sequence constructed for the proof of Theorem 50 aims to constrain one out of the total $M$ defenders at a location so as to increase the separation between the defenders. Observe that releasing multiple streams of intruders does not further increase the separation. The explanation is as follows. Suppose that two streams of intruders are released in the environment. First, note that the burst intruders must arrive after the intruders from both the streams are captured. Otherwise, the input sequence is the same as in the proof of Theorem 50. For such an input sequence there exists an online algorithm that captures intruders from one of the stream intruders. This implies that the burst never arrives and thus, the online algorithm will have a finite (may be arbitrarily high) competitive ratio.

The next result characterizes a necessary condition for the existence of online algorithms having a competitive ratio of at least 2 for when $M = 1$. This result easily extends to $M > 1$.

We first characterize locations $(t_1, \alpha_1) \in \mathcal{E}(\theta)$ and $(t_2, \alpha_2) \in \mathcal{E}(\theta)$ for the defender (Fig.

4.6), where

$$
\begin{aligned}
t_1 &= \sqrt{1 + r^2 - \tfrac{2r(1-\rho\cos(2\theta))}{\sqrt{1+\rho^2-2\rho\cos(2\theta)}}},\\
\alpha_1 &= \tan^{-1}\left(\tfrac{\sin(\theta)\sqrt{1+\rho^2-2\rho\cos(2\theta)}-r(1+\rho)\sin(\theta)}{\cos(\theta)\sqrt{1+\rho^2-2\rho\cos(2\theta)}-r(1-\rho)\cos(\theta)}\right),\\
t_2 &= \sqrt{\rho^2 + r^2 + \tfrac{2r\rho(\cos(2\theta)-\rho)}{\sqrt{1+\rho^2-2\rho\cos(2\theta)}}},\\
\alpha_2 &= \tan^{-1}\left(\tfrac{-\rho\sin(\theta)\sqrt{1+\rho^2-2\rho\cos(2\theta)}+r(1+\rho)\sin(\theta)}{\rho\cos(\theta)\sqrt{1+\rho^2-2\rho\cos(2\theta)}+r(1-\rho)\cos(\theta)}\right).
\end{aligned}
$$

The locations $(t_1, \alpha_1)$ and $(t_2, \alpha_2)$ are determined analogously to the proof of Lemma 33, and so, we only give an outline for it. Construct two circles $\mathcal{C}$ and $\mathcal{C}'$, each of radius $r$, centered at $(1, \theta)$ and $(\rho, -\theta)$ and consider a line segment $\mathcal{L}'$ that joins the centers of the two circles (Fig. 4.6). Then, location $(t_1, \alpha_1)$ (resp. $(t_2, \alpha_2)$) corresponds to the intersection points of the line segment $\mathcal{L}'$ with circles $\mathcal{C}$ and $\mathcal{C}'$, respectively.

**Theorem 36** (**Necessary condition for** $c(\mathcal{P}) \geq 2$)**.** *For any problem instance* $\mathcal{P}$ *and* $M = 1$, $c(\mathcal{P}) \geq 2$ *if*

$$
\begin{aligned}
\frac{1-\rho}{v} &\leq \sqrt{1 + \rho^2 - 2\rho\cos(2\theta)} - 2r, && \textit{if } \theta \leq \frac{\pi}{2}\\
\frac{1-\rho}{v} &\leq 1 + \rho - 2r, && \textit{if } \theta > \frac{\pi}{2}.
\end{aligned}
$$

*Proof.* The key idea is to construct input sequences for which any online algorithm is guaranteed to lose half the intruders while proving that an offline algorithm exists that can intercept all intruders. All of our input instances consists of two intruders denoted by $a$ and $b$ that arrive at location $(1, \theta)$ and $(1, -\theta)$, respectively, and we assume that the defender starts at the origin.

Two cases arise; (i) $\theta \leq \frac{\pi}{2}$ and (ii) $\theta > \frac{\pi}{2}$. We first consider case (i), i.e., $\theta \leq \frac{\pi}{2}$.

Consider that $\frac{1-\rho}{v} = \sqrt{1 + \rho^2 - 2\rho\cos(2\theta)} - 2r$ and consider an input sequence $I_1$ in which both intruders $a$ and $b$ arrive at time instant $t_1$. This is the time that the defender takes to move from the origin directly to location $(t_1, \alpha_1)$. We claim that the best way for any algorithm to capture both intruders is to capture either intruder $a$ or $b$ exactly at time

$t_1$, i.e., as soon as it arrives and then move to capture the second intruder in minimum time. The explanation is as follows.

The total time taken by the defender to capture both the intruders in the worst case is $\frac{1-x_i}{v}+\sqrt{x_i^2+\rho^2-2x_i\rho\cos(2\theta)}-2r$, where $\rho \leq x_i \leq 1$ is the radial component of the location of the first of the two intruders at the time of capture. The expression of the total time is determined as follows: The term $\frac{1-x_i}{v}$ is the intercept time for the first intruder. For the second term, construct a line segment $\mathcal{L}$ joining two points $(x_i, \theta)$ (resp. $(x_i, -\theta)$) and $(\rho, -\theta)$ (resp. $(\rho, \theta)$). Then, the length of the line segment $\mathcal{L}$ is given by $\sqrt{x_i^2+\rho^2-2x_i\rho\cos(2\theta)}$ from which we subtract $2r$ to account for the capture radius, to obtain the second term. As $\frac{1-x_i}{v}+\sqrt{x_i^2+\rho^2-2x_i\rho\cos(2\theta)}-2r$ is monotonically decreasing function of $x_i$, its minimum is achieved at $x_i = 1$. This establishes our claim that the minimum time any algorithm can take is to capture one intruder exactly when it arrives followed by the second intruder at $\sqrt{1+\rho^2-2\rho\cos(2\theta)}-2r$.

We now describe how an offline algorithm can capture both the intruders in the input sequence $I_1$. At time 0, the defender starts at the origin and moves towards location $(t_1, \alpha_1)$ capturing the intruder at location $(1, \theta)$ exactly at time $t_1$. Then the defender moves directly to location $(t_2, \alpha_2)$ exactly at time $t_1 + \sqrt{1+\rho^2-2\rho\cos(2\theta)} - 2r$ capturing the second intruder at $(\rho, -\theta)$. Note that placing the defender at $(t_1, \alpha_1)$ (resp. $(t_2, \alpha_2)$) ensures that the location $(1, \theta)$ (resp. $(\rho, -\theta)$) is on the circumference of the capture circle of the defender. Therefore, an algorithm that hopes to be better than 2-competitive must capture both the intruders in this input sequence and the only way to do so is to move to either location $(t_1, \alpha_1)$ or $(t_1, -\alpha_1)$ arriving exactly at time $t_1$.

Now consider input sequences $I_2$ and $I_3$. In $I_2$, intruder $a$ arrives at time $t_1$ and intruder $b$ arrives at time $t_1 + \epsilon$, where $\epsilon < L = 2\sin(\theta)\left(1 - \frac{r(1+\rho)}{\sqrt{1+\rho^2-2\rho\cos(2\theta)}}\right)$ and $L$ denotes the minimum time required by the defender to move from $(t_2, \alpha_2)$ to $(t_2, -\alpha_2)$. In $I_3$, intruder $b$ arrives at time $t_1$ and intruder $a$ arrives at time $t_1 + \epsilon$. Input sequence $I_2$ (resp $I_3$) are constructed for algorithms that have the defender arriving at location $(t_1, -\alpha_1)$ (resp.

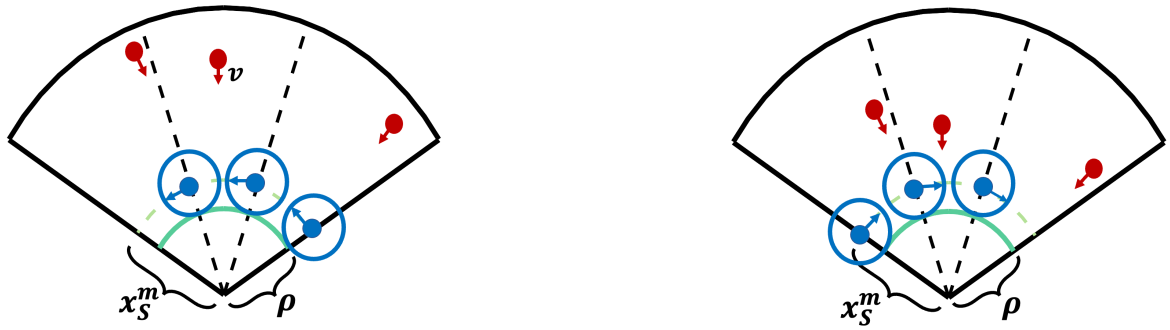
$(t_1, \alpha_1)$) at time $t_1$. Any algorithm that has the defender arriving at location $(t_1, -\alpha)$ (resp. $(t_1, \alpha_1)$) at time $t_1$ can capture only one intruder from $I_2$ (resp. ($I_3$)). As the solution is symmetric, we only provide the explanation for input sequence $I_3$. This follows as the defender can capture intruder $b$ if it moves directly to location $(t_2, \alpha_2)$ (Fig. 4.6a). However, as intruder $a$ arrives in at most $\epsilon < L$ time units, the defender will not be able to capture intruder $a$ (Fig. 4.6b). An optimal offline algorithm can capture both the intruders by simply moving to $(t_1, -\alpha_1)$ at time $t_1$, capturing intruder $a$ upon arrival and then to $(t_2, -\alpha_2)$ to capture intruder $b$.

We now consider the case when $\frac{1-\rho}{v} < \sqrt{1 + \rho^2 - 2\rho\cos(2\theta)} - 2r$. Consider input sequences $I_4$ and $I_5$. In $I_4$, intruder $a$ arrives at time $t_1$ and intruder $b$ arrives at time $t_1 + \epsilon$, where $\epsilon = \sqrt{1 + \rho^2 - 2\rho\cos(2\theta)} - 2r - \frac{1-\rho}{v}$. In $I_5$, intruder $b$ arrives at time $t_1$ and intruder $a$ arrives at time $t_1 + \epsilon$. Following similar reasoning as input instance $I_2$ and $I_3$, it follows that no online algorithm can capture both intruders from input instance $I_4$ or $I_5$.

We now consider case (ii), i.e., $\theta > \frac{\pi}{2}$. Except for when $\theta = \pi$, as the line segment $\mathcal{L}'$ will not be contained completely, the defender must move first to the origin and then to the next intercept point. Note that, the defender will do the same when $\theta = \pi$. Thus, in this case, the location $(t_1, \alpha_1)$ is $(1 - r, \theta)$ and location $(t_2, \alpha_2)$ is $(1 + \rho - 2r, -\theta)$. Following similar steps as case (i), we construct input sequences $I_1, \ldots, I_5$ (omitted for brevity) and show that no online algorithm can capture both the intruders from those input sequences.

In summary, even restricting our input sequences to $\{I_1, \ldots, I_5\}$, no online algorithm can capture both intruders whereas an optimal offline algorithm can capture both the intruders. This concludes the proof. □

We now turn our attention to design of algorithms with guaranteed upper bounds on their competitive ratios. In the next section, we design and analyze decentralized as well as cooperative algorithms, characterizing the parameter regimes in which they have provably finite competitive ratios.




(a) Defender $V_m$ is located at $(x_S^m, -(M-2m)\frac{\theta}{M})$.

(b) Defender $V_m$ is located at $(x_S^m, -(M-2m+2)\frac{\theta}{M})$.

Figure 4.7 Snapshot of Dec-Sweep for $M = 3$. The light green dashed curve denotes the angular path.

### 4.3 Algorithms

We start by defining an *angular path* for defender $V_m, 1 \leq m \leq M$. Let defender $V_m$ be located at $(x_m, \alpha_m) \in \mathcal{E}(\theta)$ for any $0 < x_m \leq 1$ and $\alpha_m \in [-\theta, \theta]$. An angular path is a circular arc centered at the origin and radius equal to the radial location of defender $V_m$, i.e., $x_m$. Formally, the angular path for defender $V_m$ is defined as $\mathcal{T}_m(x_m, \underline{\beta}_m, \overline{\beta}_m) := \{(x_m, \beta) : \underline{\beta}_m \leq \beta \leq \overline{\beta}_m\}$ for any $\underline{\beta}_m, \overline{\beta}_m \in [-\theta, \theta]$ such that $\underline{\beta}_m \leq \alpha_m \leq \overline{\beta}_m$ and $\underline{\beta}_m \neq \overline{\beta}_m$ (cf Fig. 4.7). We say that the defender completes its motion on the angular path when the defender returns to its starting location after moving along all of the points in $\mathcal{T}_m$ *twice* – once to move from the starting location $(x_m, \alpha_m)$ to $(x_m, \overline{\beta}_m)$ (resp. $(x_m, \underline{\beta}_m)$), and the second time, to move from location $(x_m, \overline{\beta}_m)$ (resp. $(x, \underline{\beta}_m)$) to location $(x_m, \underline{\beta}_m)$ (resp. $(x_m, \overline{\beta}_m)$) and then back to the starting location $(x_m, \alpha_m)$.

We start with designing decentralized algorithms followed by cooperative algorithms.

#### 4.3.1 Decentralized Algorithms

The idea for the decentralized algorithms is to first partition the environment into $M$ dominance regions and assign a single defender into each dominance region. Then, since each partition contains a single vehicle, every partition can be viewed as a conical environment with a single vehicle.

#### 4.3.1.1 Decentralized Sweep Algorithm (Dec-Sweep)

Dec-Sweep algorithm is an open loop as well as memoryless algorithm and is described as follows.

Dec-Sweep partitions the environment $\mathcal{E}(\theta)$ into $M$ dominance regions, each of angle $\frac{2\theta}{M}$. Each defender $V_m, 1 \leq m \leq M$ starts at location $(x_S^m, -(M-2m+1)\frac{\theta}{M})$, where

$$x_S^m \in [\rho - r, \min\{1-r, \rho+r\}].$$

The choice of $x_S^m$ will be justified shortly. Each defender $V_m, 1 \leq m \leq M$, then follows the angular path $\mathcal{T}_m(x_S^m, -(M-2m+2)\frac{\theta}{M}, -(M-2m)\frac{\theta}{M})$ moving towards location $(x_S^m, -(M-2m)\frac{\theta}{M})$ at time 0. More precisely, at time 0, each defender $V_m, 1 \leq m \leq M$, picks a velocity with unit magnitude and direction tangent to the angular path $\mathcal{T}_m$, oriented to the right until it reaches $(x_S^m, -(M-2m)\frac{\theta}{M})$. Note that the angular coordinate $-(M-2m)\frac{\theta}{M}$ and $-(M-2m+2)\frac{\theta}{M}$ correspond to the angular coordinates of the two endpoints of $\partial R_m$. Once it reaches $(x_S^m, -(M-2m)\frac{\theta}{M})$ (cf Fig. 4.7a), the defender switches direction and moves towards the other endpoint of $\partial R_m$, i.e., $(x_S^m, -(M-2m+2)\frac{\theta}{M})$ (cf Fig. 4.7b). From this moment on, the defender only switches direction after it reaches an endpoint of $\partial R_m$. More precisely, upon reaching location $(x_S^m, -(M-2m)\frac{\theta}{M})$, the defender keeps on moving on an angular path $\mathcal{T}_m(x_S^m, -(M-2m)\frac{\theta}{M}), -(M-2m+2)\frac{\theta}{M}$.

The following three aspects jointly justify the choice of location $x_S^m$, $\forall m \in \{1, \dots, M\}$.

1. There is no benefit for defender $V_m$ to be located below a distance of $\rho - r$ from the origin as the capture circle will be completely inside the perimeter and the defender $V_m$ cannot capture any intruder using the angular sweep algorithm. This yields $x_S^m \geq \rho - r$.

2. For any radial location $x_S^m > \rho + r$, the defender $V_m$ will take exactly $4\frac{\theta}{M}x_S^m$ time units to complete one angular path. On the other hand, in the worst case, the intruders will require exactly $\frac{2r}{v}$ time to cross the region swept by the defender (cf. Figure 4.7). Since the time taken $4\frac{\theta}{M}x_S^m$ by defender $V_m$ increases with $x_S^m$, while the time taken

$\frac{2r}{v}$ by the intruders remains the same for any $x_S^m > \rho + r$, it follows that there is no benefit for $x_S^m > \rho + r$. Equivalently, $x_S^m \leq \rho + r$ must hold.

3. Finally, to ensure that $x_S^m + r$ is contained within the environment, we require $x_S^m \leq 1 - r$.

Since $\rho - r < \rho + r$ and $\rho - r < 1 - r$, $x_S^m$ is well defined.

**Theorem 37.** *Algorithm Decentralized Sweep is* 1*-competitive, i.e.,* $c_{\text{Dec-Sweep}} = 1$*, for any problem instance* $\mathcal{P}$ *which satisfies*

$$v \leq \min_{m \in \{1,\ldots,M\}} \Big\{ \frac{M(x_S^m + r - \rho)}{4\theta x_S^m} \Big\},$$

*for any* $x_S^m \in \Big[\rho - r, \min\{1 - r, \rho + r\}\Big]$ *and* $\forall m \in \{1, \ldots, M\}$*. Otherwise, Decentralized Sweep is not c-competitive for any constant c.*

*Proof.* Without loss of generality, we assume that defender $V_m, 1 \leq m \leq M$, has just left location $(x_S^m, -(M-2m)\frac{\theta}{M})$ at time instant $t$ and an intruder $i$ is located at $(x_S^m + r, -(M - 2m)\frac{\theta}{M})$. Defender $V_m$ takes exactly $4\frac{\theta}{M}x_S^m$ time units to reach location $(x_S^m + r, -(M - 2m)\frac{\theta}{M})$ whereas intruder $i$ takes exactly $\frac{x_S^m + r - \rho}{v}$ time units to reach the perimeter. Given the condition $v \leq \frac{M(x_S^m + r - \rho)}{4\theta x_S^m}$ which is equivalent to $4\frac{\theta}{M}x_S^m \leq \frac{x_S^m + r - \rho}{v}$, it follows that intruder $i$ will not be lost. Since this explanation holds for every defender $V_m, 1 \leq m \leq M$, the first result is established. We now establish the second result. Without loss of generality consider that $v > \frac{M(x_S^M + r - \rho)}{4\theta x_S^M}$. Then we can construct an input sequence with intruders only arriving at $(1, \theta)$ such that at the time instant defender $V_M$ leaves location $(x_S^M, \theta)$, arbitrary number of intruders are located at location $(x_S^M + r, \theta)$ at the same time instant. Since, $v > \frac{M(x_S^M + r - \rho)}{4\theta x_S^M}$, all intruders will be lost and the result follows. $\square$

**Corollary 4.** *Placing each defender* $V_m$ *at a radial distance of* $\min\{1 - r, \rho + r\}$ *maximizes the speed ratio for which Theorem 37 holds. In other words,* $c_{\text{Dec-Sweep}} = 1$ *if* $v \leq \min\{\frac{2rM}{4\theta(\rho + r)}, \frac{M(1-\rho)}{4\theta(1-r)}\}$.

This corollary follows from the fact that $\frac{M(x_S^m+r-\rho)}{4\theta x_S^m}$ is a monotonically increasing function of $x_S^m$. We now establish that any modification to the partitioning in Dec-Sweep Algorithm does not improve the parameter space. Note that since Dec-Sweep is already 1-competitive, one cannot further improve the competitive ratio.

**Corollary 5.** *Partitioning the environment into dominance regions of equal angles maximizes the speed ratio for which Theorem 37.*

*Proof.* We establish this claim by arriving at a contradiction. Suppose that there exist two dominance regions that have different angles $2\gamma_1$ and $2\gamma_2$ such that $2\gamma_1 < 2\gamma_2$.

Now, all defenders must be placed so that they take the same time to complete a sweep of their respective dominance regions. Otherwise, the inputs will be concentrated in the dominance region of the slowest such defender, which will govern the upper bound on $v$ for 1-competitiveness. Therefore, we must have

$$4\gamma_2 x_S^2 = 4\gamma_1 x_S^1 \Rightarrow x_S^1 = x_S^2 \frac{\gamma_2}{\gamma_1},$$

where $x_S^1$ and $x_S^2$ denote the radial locations of the corresponding defenders in the two dominance regions under consideration. Thus, in the worst-case, for 1-competitiveness we require

$$v \leq \min\left\{\frac{x_S^2 + r - \rho}{4\gamma_2 x_S^2}, \frac{x_S^1 + r - \rho}{4\gamma_1 x_S^1}\right\} = \frac{x_S^2 + r - \rho}{4\gamma_2 x_S^2}.$$

From Corollary 4 and since $x_S^2 < x_S^1$, it follows that $x_S^2 < \min\{1 - r, \rho - r\}$ as $x_S^1 = \min\{1-r, \rho-r\}$. In other words, the requirement on $v$ for 1-competitiveness can be relaxed by increasing $\gamma_1$ until $\gamma_1 = \gamma_2$. This concludes the proof. □

Although Dec-Sweep is a 1-competitive algorithm, it is effective only for small values of speed ratio $v$. Thus, the motivation behind our second algorithm is to capture intruders with higher speeds at the expense of losing some intruders.

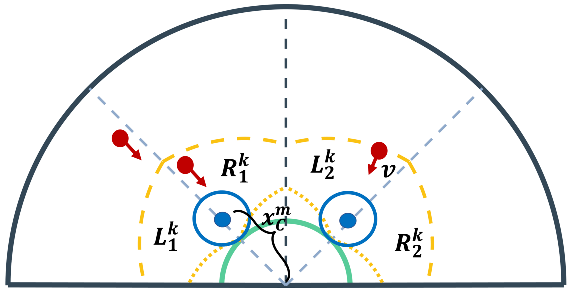



Figure 4.8 Description of Dec-CaC Algorithm for $M = 2$. The black dashed line depicts partition of the environment. The region between the yellow-dashed and yellow-dot lines denote the set $R_m^k$ and $L_m^k$. $|R_1^k| = |L_2^k| = 1$ and $|L_1^k| = |R_2^k| = 0$.

#### 4.3.1.2 Decentralized Compare and Capture (Dec-CaC)

The intuition behind this algorithm is to move the defender within a portion of its assigned dominance region so as to obtain a higher parameter regime. In other words, the algorithm first partitions the environment into $M$ dominance regions and positions a single defender at the center of each dominance region. The algorithm then compares the number of intruders arriving on each side of every defender. After comparison, the defenders in every dominance region move on an angular path towards the side which has higher number of intruders. Hence, our second algorithm is not open-loop. However, we will see that this is a memoryless algorithm when we formally describe the algorithm.

We first provide some definitions that will be useful in describing and analyzing the Dec-CaC algorithm. An epoch $k_m$ in a dominance region $m$ is defined as the time interval in which defender $V_m$ completes its motion on angular path $\mathcal{T}_m(x_C^m, \underline{\beta}_m, \overline{\beta}_m)$ with a specified distance $x_C^m$ such that

$$x_C^m \in \left[\rho - r, \min\{\rho + r, \frac{M(1-r)}{M + v\theta}\}\right].$$

The distance $x_C^m, \forall m \in \{1, \dots, M\}$, for defender $V_m$ in dominance region $m$ is same in all epochs whereas $\underline{\beta}_m$ and $\overline{\beta}_m$ are set at the start of every epoch $k$ for every dominance region $m$. The choice of $x_C^m$ and the values that $\underline{\beta}_m$ and $\overline{\beta}_m$ take will be described shortly. Denote $|R_m^k|$ and $|L_m^k|$ as the total number of intruders contained in the sets $R_m^k$ and $L_m^k$ (cf. Figure

**Algorithm 5:** Decentralized Compare-and-Capture (Dec-CaC) Algorithm

**for** *each defender* $1 \leq m \leq M$ **do**
  Select $x_C \in [\rho - r, \min\{\rho + r, \frac{M(1-r)}{M+v\theta}]\}$.
  **for** *each epoch* $k \geq 1$ **do**
    **if** $|L_m^k| \leq |R_m^k|$ **then**
      Set $\underline{\beta}_m = -(M - 2m + 1)\frac{\theta}{M}$, $\overline{\beta}_m = -(M - 2m)\frac{\theta}{M}$
      Move on angular path to location $(x_C^m, \overline{\beta}_m)$.
      Move on angular path to return to location $(x_C^m, \underline{\beta}_m)$.
    **else**
      Set $\underline{\beta}_m = -(M - 2m + 2)\frac{\theta}{M}$, $\overline{\beta}_m = -(M - 2m + 1)\frac{\theta}{M}$
      Move on angular path to location $(x_C^m, \underline{\beta}_m)$
      Move on angular path to return to location $(x_C, \overline{\beta}_m)$.
    **end**
  **end**
**end**

4.8), respectively, defined as

$$
\begin{aligned}
R_m^k :=& \{(y, \phi) : \rho + \phi x_C^m v < y \leq \\
& \min\{1, x_C^m + r + (2\tfrac{\theta}{M} - \phi) v x_C^m\} \forall \phi \in [0, \tfrac{\theta}{M}] \text{ and} \\
L_m^k :=& \{(y, \phi) : \rho - \phi x_C^m v < y \leq \\
& \min\{1, x_C^m + r + (2\tfrac{\theta}{M} + \phi) v x_C^m\} \forall \phi \in [-\tfrac{\theta}{M}, 0).
\end{aligned}
$$

Dec-CaC algorithm is defined in Algorithm 5 and is summarized as follows. At the start of every epoch $k_m$, each defender $V_m$ compares the total number of intruders in the set $R_m^k$ and $L_m^k$. If $|R_m^k| < |L_m^k|$, then defender $V_m$ moves on the angular path, $\mathcal{T}_m(x_C^m, \underline{\beta}_m, \overline{\beta}_m)$ with $\underline{\beta}_m = -(M - 2m + 2)\frac{\theta}{M}$ and $\overline{\beta}_m = -(M - 2m + 1)\frac{\theta}{M}$. Otherwise $\underline{\beta}_m = -(M - 2m + 1)\frac{\theta}{M}$ and $\overline{\beta}_m = -(M - 2m)\frac{\theta}{M}$.

For the initial case, we assume time 0 as the time when the first intruder is released. Each defender $V_m$ starts at location $(x_C^m, -(M - 2m + 1)\frac{\theta}{M})$ and begins its first epoch at time 0.

We now justify the range of values for $x_C^m$. The explanation for $x_C^m \geq \rho - r$ and $x_C^m \leq \rho + r$ is analogous to that for the choice of $x_S^m$ (points 1 and 2) for the Dec-Sweep algorithm.

Further, we require $x_C^m + r + vx_C^m \frac{\theta}{M} \leq 1 \Rightarrow x_C^m \leq \frac{M(1-r)}{M+v\theta}$ for the algorithm to have a finite competitive ratio. This yields

$$x_C^m \in \Big[\rho - r, \min\Big\{\rho + r, \frac{M(1-r)}{M+v\theta}\Big\}\Big].$$

Note that $x_C^m$ is well defined only if

$$\rho - r \leq \frac{M(1-r)}{M+v\theta} \Rightarrow v \leq \frac{M(1-\rho)}{\theta(\rho - r)}. \tag{4.3}$$

Recall that once a value of $x_C^m$ is selected for a defender $V_m$, it remains fixed for all epochs and two defenders may not have the same value of $x_C^m$. We now establish two results that will be useful in establishing that Algorithm Dec-CaC is 2-competitive.

**Lemma 38.** *Any intruder that lies beyond*[3] *the location* $(x_C^m + r + (2\frac{\theta}{M} - \phi)vx_C^m, \phi)$, $\forall \phi \in [-(M-2m+1)\frac{\theta}{M}, -(M-2m)\frac{\theta}{M}]$ *in epoch* $k_m$ *and in a dominance region* $m$*, for any* $1 \leq m \leq M$*, will either be contained in the set* $R_m^{k+1}$ *or in* $L_m^{k+1}$ *in epoch* $k_m + 1$ *and is not lost at the start of epoch* $k_m + 1$ *if*

$$v \leq \frac{M(x_C^m + r - \rho)}{2\theta x_C^m}.$$

*Proof.* Without loss of generality, assume that $|L_m^k| < |R_m^k|$ at epoch $k_m$ holds in dominance region $m \in \{1, \ldots, M\}$. The total time taken by defender $V_m$ to capture intruders in $R_m^k$ and return back to its starting location $(x_C^m, -(M-2m+1)\frac{\theta}{M})$ is $2\frac{\theta}{M}x_C^m$. In the worst-case, in order for any intruder $i$ to be not considered in the start of epoch $k_m$, the intruder $i$ must be located just above $(x_C^m + r + \frac{\theta}{M}vx_C^m, -(M-2m+2)\frac{\theta}{M})$ at the start of epoch $k_m$. By the time the defender reaches location $(x_C^m, -(M-2m+2)\frac{\theta}{M})$, intruder $i$ will be located just above the location $(x_C^m + r, -(M+2m)\theta)$ and will not be captured.

If $v \leq \frac{M(x_C^m + r - \rho)}{2\theta x_C^m}$ holds for dominance region $m$, intruder $i$ will be at least $\frac{\theta}{M}vx_C^m$ distance away from the perimeter at the end of epoch $k_m$. Thus, intruder $i$ will be considered in the start of epoch $k_m + 1$ as per the definitions of the sets $L_m^{k+1}$ and $R_m^{k+1}$. Clearly, as the intruder

[3] intruders with radial coordinate more than $x_C^m + r + (2\frac{\theta}{M} - \phi)vx_C^m$

$i$ will be considered for comparison in epoch $k_m + 1$, it is not lost unless the defender decides to move to $L_m^{k+1}$ in epoch $k+1$. This concludes the proof. □

**Theorem 39** (Dec-CaC competitiveness)**.** *For any problem instance $\mathcal{P}$, suppose that*

$$v \leq \min_{m \in \{1,\ldots,M\}} \Big\{ \frac{M(x_C^m + r - \rho)}{2\theta x_C^m} \Big\} \tag{4.4}$$

*and equation* (4.3) *holds. Then, for all $m \in \{1,\ldots,M\}$, with the choice of an*

$$x_C^m \in \Big[\rho - r, \min\{\rho + r, \frac{M(1-r)}{M + v\theta}\}\Big],$$

*$c_{Dec-CaC} = 2$, i.e., Algorithm Dec-CaC algorithm is 2-competitive.*

*Proof.* The idea behind this proof is to show that in any dominance region $1 \leq m \leq M$, defender $V_m$ always captures at least the same number of intruders it loses. The result then follows as the above claim then holds for every dominance region. Lemma 38 ensures that every intruder will belong to either set $L_m^k$ or $R_m^k$ in every epoch $k_m$. In every epoch $k_m$, defender $V_m$ compares the total number of intruders on either side contained in the set $L_m^k$ and $R_m^k$ and moves to the side where the number of intruders is higher. Thus, it is guaranteed that the defender will capture at least half of the total number of intruders that arrive in the environment, assuming that an optimal offline algorithm can capture all intruders. □

**Remark 3.** *As $\frac{M(x_C^m + r - \rho)}{2\theta x_C^m}$ is a monotonically increasing function of $x_C^m$, positioning each defender $V_m, 1 \leq m \leq M$ at $\min\{\rho + r, \frac{M(1-r)}{M+v\theta}\}$ yields $v \leq \min\{\frac{Mr}{\theta(\rho+r)}, \frac{M(1-\rho)}{(2-3r+\rho)\theta}, \frac{M(1-\rho)}{\theta(\rho-r)}\}$.*

Similar to Dec-Sweep Algorithm, the next result establishes that there is no benefit obtained in the parameter regime of Dec-CaC algorithm if the environment is not partitioned into equal dominance region. As the proof is analogous to that of Corollary 4, we omit the proof for brevity.

**Corollary 6.** *Partitioning the environment into equal dominance region of equal angles yields the maximum speed ratio for which Dec-CaC Algorithm is 2-competitive.*

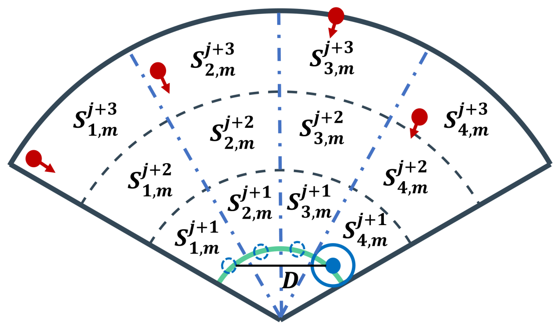



Figure 4.9 Description of Dec-SNP Algorithm for $m$-th dominance region and $n_s = 4$. The three intervals of length $Dv$ are represented by the black dashed curves and the resting point for defender $V_m$ are denoted by blue dashed circles. The defender $V_m$ is located at $(x_4^m, \alpha_4^m)$.

As explained in more detail in Section 4.4, Dec-Sweep and Dec-CaC are not very effective for small values of capture radius $r$ and high values of perimeter size $\rho$. Our next algorithm addresses this issue by further dividing each of the $M$ partition into sectors.

#### 4.3.1.3 Decentralized Stay Near Perimeter Algorithm (Dec-SNP)

Unlike the previous two algorithms, in this algorithm, the defenders do not follow an angular path. Instead, the idea is to further divide each of the $M$ dominance regions into sectors and position the defenders close to the perimeter in a specific sector of the dominance region.

In algorithm Dec-SNP, we divide each dominance region $m, 1 \leq m \leq M$, into $n_s = \lceil \frac{\theta}{M\theta_s} \rceil$ sectors, where $2\theta_s = 2\arctan(\frac{r}{\rho})$ denotes the angle of each sector. Let $N_{l,m}, l \in \{1, \ldots, n_s\}$, denote the $l$-th sector of the $m$-th dominance region, where $N_{1,m}$ denotes the leftmost sector of the $m$-th dominance region (cf. Fig. 4.9). Further, let $\partial R_l^m$ denote the portion of the perimeter contained in sector $N_{l,m}$. Then, a resting point $(x_l^m, \alpha_l^m) \in \mathcal{E}(\theta)$ of sector $N_{l,m}$ is defined as a location for defender $V_m$ such that when positioned at that location, all of $\partial R_l^m$ is contained completely within the capture radius of the defender. Mathematically,

$$(x_l^m, \alpha_l^m) := \left(\frac{\rho}{\cos(\theta_s)}, -(M - 2m + 2)\frac{\theta}{M} + (2l - 1)\theta_s\right).$$

Lastly, we let $D$ denote the distance between the two resting points that are furthest apart

**Algorithm 6:** Decentralized Stay Near Perimeter (Dec-SNP) Algorithm

1 **for** *each defender* $1 \leq m \leq M$ **do**
2     Stay at $(x_1^m, \alpha_1^m)$ until time $2D$.
3     **for** *each* $j \geq 1$ **do**
4         $k^* = \arg\max_{k \in \{1,\ldots,n_s\}} \{\eta_i^1(m), \ldots, \eta_i^{n_s}(m)\}$
5         $N_{o,m} = N_{k^*,m}$
6         **if** $N_{o,m} \neq N_{i,m}$ *and* $|S_{o,m}^{j+2}| \geq |S_{i,m}^{j+1}|$ **then**
7             Move to $(x_o^m, \alpha_o^m)$ and then capture $|S_{o,m}^{j+2}|$
8         **else**
9             Stay at $(x_i^m, \alpha_i^m)$ and capture $|S_{i,m}^{j+1}|$
10         **end**
11     **end**
12 **end**

in a dominance region (cf. Fig. 4.9). Formally,

$$D = \begin{cases} 2\frac{\rho}{\cos(\theta_s)}\sin((n_s-1)\theta_s), & \text{if } (n_s-1)\theta_s < \frac{\pi}{2} \\ 2\frac{\rho}{\cos(\theta_s)}, & \text{otherwise.} \end{cases} \tag{4.5}$$

Note that each of the $M$ dominance regions will have equal number of $n_s$ sectors and therefore, the same value of $D$. After dividing each dominance region $m$ in the environment into $n_s$ sectors, Dec-SNP algorithm radially divides the environment $\mathcal{E}(\theta)$ into three intervals of length $Dv$, corresponding to time intervals of time length $D$ each. Specifically, the $j^{th}$ time interval for any $j > 0$ is defined as the time interval $[(j-1)D, jD]$. In order to ensure a finite competitiveness for this algorithm, we require $\frac{1-\rho}{v} \geq 3D$, i.e., the intruders require at least $3D$ time units to reach the perimeter after they are released in the environment. For any $j \geq 1$, let $S_{l,m}^j$ be the set of intruders present in sector $N_{l,m}$ in the $j^{th}$ time interval.

The Dec-SNP algorithm (cf. Algorithm 6) is based on the following two steps. Each defender $V_m, 1 \leq m \leq M$, first selects the sector (within dominance region $m$) with the maximum number of intruders, and then determines whether it is beneficial to move to the resting location of that sector. These two steps are achieved by the two comparisons **C1** (lines 7-8 in Algorithm 6) and **C2** (lines 9-13 in Algorithm 6) elaborated below.

**Comparison C1**: For each dominance region $1 \leq m \leq M$, Dec-SNP determines that sector which has the most number of intruders in the last two intervals as compared to the total number of intruders in the entire sector in which the defender $V_m$ is located. In particular, suppose that defender $V_m$ is located at the resting point of sector $N_{i,m}, 1 \leq i \leq n_s$ at the $j^{th}$ iteration. Corresponding to any sector $N_{l,m}, 1 \leq l \leq n_s$ in dominance region $m$, we define $\eta_i^l(m)$ as

$$\eta_i^l(m) = \begin{cases} |S_{l,m}^{j+2}| + |S_{l,m}^{j+3}|, \text{ if } l \neq i \\ |S_{i,m}^{j+1}| + |S_{i,m}^{j+2}| + |S_{i,m}^{j+3}|, \text{ if } l = i \end{cases}$$

Then, Dec-SNP selects the sector $N_{k^*,m}$ for each dominance region $m$, where

$$k^* = \arg \max_{k \in \{1, \ldots n_s\}} \{\eta_i^1(m), \ldots, \eta_i^{n_s}(m)\}.$$

If there are multiple sectors in dominance region $m, m \in \{1, \ldots, M\}$ with same number of intruders, then Dec-SNP breaks the tie as follows. If the tie includes the sector $N_{i,m}$, then Dec-SNP selects $N_{i,m}$. Otherwise, Dec-SNP selects the sector with the maximum number of intruders in the interval $j+2$. If this results in another tie, then this second tie can be resolved by picking the sector with the least index. Let the sector chosen for dominance region $m$ as the outcome of comparison **C1** be $N_{o,m}, o \in \{1, \ldots, n_s\}$.

**Comparison C2**: If the sector chosen is $N_{o,m}, o \neq i$, and the total number of intruders in the set $S_{o,m}^{j+2}$ is greater than or equal to the total number of intruders in the set $S_{i,m}^{j+1}$, then Dec-SNP moves defender $V_m$ to location $(x_o^m, \alpha_o^m)$ arriving in at most $D$ time units. Then, defender $V_m$ waits at that location for another $D$ time units to capture all intruders in $S_{o,m}^{j+2}$. Otherwise (i.e., if $o = i$ or $|S_{o,m}^{j+2}| < |S_{i,m}^{j+1}|$), defender $V_m$ stays at its current location $(x_i^m, \alpha_i^m)$ for $D$ time units and captures $S_{i,m}^{j+1}$. After capture of either $S_{o,m}^{j+2}$ or $S_{i,m}^{j+1}$, the defender then reevaluates at time $D$ or $2D$, respectively.

At time 0, each defender $V_m, 1 \leq m \leq M$ waits at the resting point of the first sector of their corresponding dominance regions, i.e., the resting point of $N_{1,m}$, for $2D$ time units. The first epoch begins at time $2D$.

**Lemma 40.** *For each $m \in \{1, \ldots, M\}$, let defender $V_m$ be located at $(x_i^m, \alpha_i^m)$ of a sector $N_{i,m}, 1 \le i \le n_s$. Then, for any $j \ge 1$, $V_m$ always captures either $S_{i,m}^{j+1}$ or $S_{o,m}^{j+2}$, where $S_{o,m}^{j+2}$ denotes the set of intruders in sector $N_{o,m}$ selected after **C1**.*

*Proof.* Suppose that after comparison **C1**, $N_{o,m} = N_{i,m}$. Then the result follows from Algorithm 6 (line 9) because the defender stays at its current location, i.e., $(x_i^m, \alpha_i^m)$ and captures $S_{i,m}^{j+1}$.

Now suppose that after comparison **C1**, $N_{o,m} \neq N_{i,m}$. Then there are two cases: (i) Either the defender decides to stay at its current location for $D$ time interval implying that $|S_{i,m}^{j+1}| > |S_{o,m}^{j+2}|$ (line 9) or (ii) the defender decides to move to the resting point corresponding to the sector $N_{o,m}$ selected from **C1** implying that $|S_{i,m}^{j+1}| \le |S_{o,m}^{j+2}|$ (line 7). In case (i), the defender stays at its current location and captures $S_{i,m}^{j+1}$. In case (ii), the defender spends at most $D$ time units to moves to the resting point of the sector $N_{o,m}$ and then captures intruders in the set $S_{o,m}^{j+2}$. This concludes the proof. □

To establish the competitive ratio of Algorithm Dec-SNP, we use an accounting analysis in which captured intervals *pay* for the lost intervals or equivalently, captured intervals are *charged* for the intervals lost. The following two lemmas, proofs of which are contained in Appendix, will jointly establish the competitive ratio of Dec-SNP algorithm. A single *charge* to a particular captured interval consists of all the lost intervals that were considered in a comparison step **C1** involving that particular captured interval. Note that we consider the first interval of every sector in which the defender is not contained as a lost interval because that interval can never be captured by that defender.

**Lemma 41.** *Using Algorithm Dec-SNP, for every defender $V_m, m \in \{1, \ldots, M\}$, any* two consecutive *captured intervals pay for a total of* $3(n_s - 1)$ *lost intervals.*

*Proof.* Please see Appendix (12) □

We now establish that each lost interval is fully accounted for by the captured intervals.

**Lemma 42.** *Each lost interval is accounted for by the captured intervals in the Dec-SNP algorithm.*

*Proof.* Please see Appendix (12) □

Together, these two lemmas lead to the following main result.

**Theorem 43** (Dec-SNP competitiveness)**.** *For any problem instance $\mathcal{P}$ that satisfies $3D \leq \frac{1-\rho}{v}$, $c_{Dec\text{-}SNP} = \frac{3n_s-1}{2}$, where $n_s = \lceil \theta/(M\theta_s) \rceil$, $\theta_s = \arctan(r/\rho)$ and $D$ is defined in* (4.5).

*Proof.* From Lemma 41 and Lemma 42 it follows that for any given trace of SNP algorithm, every two consecutively captured intervals pay for $3n_s - 3$ lost intervals and every lost interval is accounted by two consecutive captured intervals. Thus, the claim follows. □

In all of the above algorithms, the central idea was to partition the environment $\mathcal{E}(\theta)$ into $M$ dominance regions and then assign a single defender to each dominance region. Next, we design two cooperative algorithms with provably finite competitive ratio.

### 4.3.2 Cooperative Algorithms

#### 4.3.2.1 Cooperative Compare and Capture Algorithm (Coop-CaC)

Recall that in Algorithm Dec-CaC, every defender captures the intruders from only one out of the two sets, $R_m^k$ and $L_m^k$. The idea behind Coop-CaC algorithm is to have an overlap of the set $R_m^k$ and $L_{m+1}^k$ of the dominance region $m$ and $m+1$ so that the defenders do not lose the same number of intruders that they capture. Since there is an overlap of the sets, we require that $M$ is an even number for Algorithm Coop-CaC.

Algorithm Coop-CaC first partitions the environment into $\frac{M}{2}$ dominance regions, each of angle $\frac{4\theta}{M}$ and assigns two defenders to each dominance region $m \in \{1, \ldots, \frac{M}{2}\}$. Each dominance region is then further divided into three equal sectors, each of angle $\frac{4\theta}{3M}$ (Fig. 4.10). As will be described shortly, the motion of the defenders in any dominance region $m \in \{1, \ldots, \frac{M}{2}\}$ is independent of the motion of the defenders in dominance region $m'$, $\forall m' \in \{1, \ldots, \frac{M}{2}\} \setminus \{m\}$. However, the two defenders within dominance region $m$ cooperate

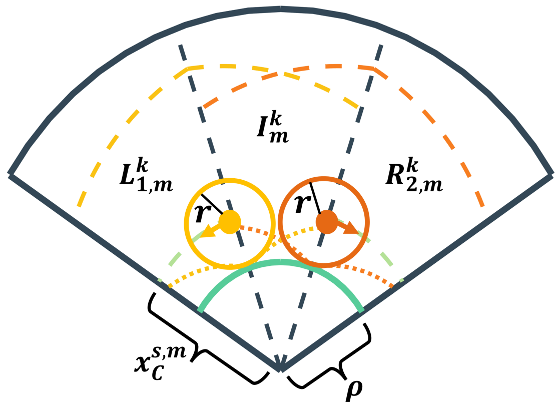



Figure 4.10 Description of Coop-CaC for a dominance region $m$ with three sectors. The two defenders (shown in yellow and orange color) share the sector in the middle. The region between the yellow dashed lines and the yellow dots and on the left (resp. right) of the first (moving clockwise) black dashed line is the set $L_{1,m}^k$ (resp. $R_{1,m}^k$). The region between the orange dashed lines and the orange dots and on the right (resp. left) of the second black dashed line is the set $R_{2,m}^k$ (resp. $L_{2,m}^k$). The region between the intersection of the yellow and orange dashed lines and dots denotes the set $I_m^k = R_{1,m}^k \cap L_{2,m}^k$. The green dashed line denotes the angular paths for the two defenders when $|L_{1,m}^k| > |I_m^k|$ and $|R_{2,m}^k| > |I_m^k|$ holds.

with each other. The cooperation is achieved by assigning the two defenders a common sector. In particular, let $V_s^m$ denote the defender assigned to sector $s$ and $s+1$ in dominance region $m$, where $s \in \{1, 2\}$. Then the two defenders $V_1^m$ and $V_2^m$ share the second sector of the $m$-th dominance region (cf. Fig. 4.10).

Before we summarize the algorithm, we first determine the values of the parameters $x_C^{s,m}, \underline{\beta}_{s,m}$ and $\overline{\beta}_{s,m}$ which characterize the angular paths $\mathcal{T}_s(x_C^{s,m}, \underline{\beta}_{s,m}, \overline{\beta}_{s,m})$ of defender $V_s^m$ for all $1 \leq s \leq 2$ in dominance region $m$. Let an epoch $k_m$ of dominance region $m$ be defined as the time the two defenders in dominance region $m$ take to complete their motion on their respective angular paths. Note that this requires that the time taken by the two defenders must be the same. Since the angle of each of the sector in a dominance region $m$ is equal, it follows that $x_C^{1,m} = x_C^{2,m}$ must hold in every dominance region $m$. The distance $x_C^{s,m}$ is fixed for a defender in all epochs. However, unlike Algorithm Dec-CaC, $\underline{\beta}_{s,m}$ and $\overline{\beta}_{s,m}$ are set at the start of every epoch $k$ for every sector $s$ based on the number of intruders as well as the angular path of the other defender in the same dominance region $m$.

We now determine the range of values that the parameter $x_C^{s,m}$ can take. From points 1 and 2 of the justification for $x_S^m$ in Algorithm Dec-Sweep, it follows that $x_C^{s,m} \leq \rho + r$ and $x_C^{s,m} \geq \rho - r$. Further, as we require that $x_C^{s,m} + r + \frac{4\theta v x_C^{s,m}}{3M} \leq 1$, we obtain $x_C^{s,m} \leq \frac{3M(1-r)}{3M+4\theta v}$. Thus,

$$x_C^{s,m} \in \Big[\rho - r, \min\{\rho + r, \tfrac{3M(1-r)}{3M+4\theta v}\}\Big].$$

Note that the set $x_C^{s,m}$ is defined when

$$\rho - r < \frac{3M(1-r)}{3M+4\theta v} \Rightarrow v < \frac{3M(1-\rho)}{4\theta(\rho - r)}. \tag{4.6}$$

For all $s \in \{1, 2\}$ and every dominance region $m \in \{1, \ldots, \frac{M}{2}\}$, denote $|R_{s,m}^k|$ and $|L_{s,m}^k|$ as the total number of intruders contained in the sets $R_{s,m}^k$ and $L_{s,m}^k$ (cf. Figure 4.10), respectively. Mathematically, the sets are defined as

$$\begin{aligned}
&R_{s,m}^k := \{(y,\phi) : \rho + \phi x_C^{s,m} v < y \leq \\
&\min\{1, x_C^{s,m} + r + (\tfrac{8\theta}{3M} - \phi) v x_C^{s,m}\} \forall \phi \in [0, \tfrac{4\theta}{3M}] \\
&L_{s,m}^k := \{(y,\phi) : \rho - \phi x_C^{s,m} v < y \leq \\
&\min\{1, x_C^{s,m} + r + (\tfrac{8\theta}{3M} + \phi) v x_C^{s,m}\} \forall \phi \in [-\tfrac{4\theta}{3M}, 0).
\end{aligned}$$

Note that $|R_{1,m}^k|$ may not be equal to $|L_{2,m}^k|$ for any $m \in \{1, \ldots, \frac{M}{2}\}$. Finally, let $|I_m^k|$ denote the number of intruders contained in the intersection of the set $I_m^k \triangleq R_{1,m}^k \cap L_{2,m}^k$( cf Fig. 4.10). Then, Algorithm Coop-CaC which is defined in Algorithm 7, is summarized as follows.

For every dominance region $m$, Algorithm Coop-CaC computes the total number of intruders in the set $L_{1,m}^k$, $R_{2,m}^k$ and $I_m^k$. As that there are three sectors and two defenders in a particular dominance region, the idea is to capture the two out of the three sets that contain the most number of intruders. This requires that the two defenders must be assigned the appropriate sets. Once the set is assigned, the defenders can then capture the intruders by moving on the angular paths specified by the sets assigned to them. Thus, in what follows, we describe the assignment of the sets to the defenders, which can be summarized into two cases.

**Case 1:** Both $|I_m^k| \geq |R_{2,m}^k|$ and $|I_m^k| \geq |L_{1,m}^k|$ holds. This means that the set $I_m^k$ has the most number of intruders. Thus, by determining which out of the remaining two sets, i.e., $R_{2,m}^k$ and $L_{1,m}^k$, has more number of intruders, Algorithm Coop-CaC assigns the sets to the two defenders. Mathematically, if both $|I_m^k| \geq |R_{2,m}^k|$ and $|I_m^k| \geq |L_{1,m}^k|$ hold, then

- If $|L_{1,m}^k| \geq |R_{2,m}^k|$, then defender $V_1^m$ and $V_2^m$ moves on the angular paths $\mathcal{T}_1(x_C^{s,m}, \underline{\beta}_{1,m}, \overline{\beta}_{1,m})$ and $\mathcal{T}_2(x_C^{s,m}, \underline{\beta}_{2,m}, \overline{\beta}_{2,m})$, respectively, with

  $$\underline{\beta}_{s,m} = \phi_1 - \tfrac{4\theta}{3M}, \overline{\beta}_{s,m} = \phi_1,\ \forall s \in \{1, 2\},$$

  where $\phi_1 = -(\frac{3}{2}(M - 4(m-1)) - 2s)\frac{2\theta}{3M}$ and is measured from the positive $y$-axis. In other words, both defenders move counter clockwise by moving on their respective angular paths.

- Otherwise, i.e., $|L_{1,m}^k| < |R_{2,m}^k|$ holds then defenders $V_1^m$ and $V_2^m$ move on the angular paths $\mathcal{T}_1(x_C^{s,m}, \underline{\beta}_{1,m}, \overline{\beta}_{1,m})$ and $\mathcal{T}_2(x_C^{s,m}, \underline{\beta}_{2,m}, \overline{\beta}_{2,m})$, respectively, with

  $$\underline{\beta}_{s,m} = \phi_1, \overline{\beta}_{s,m} = \phi_1 + \tfrac{4\theta}{3M},\ \forall s \in \{1, 2\}.$$

  This means that both defenders move clockwise by moving on their respective angular paths.

**Case 2:** Both $|I_m^k| \geq |R_{2,m}^k|$ and $|I_m^k| \geq |L_{1,m}^k|$ do not hold. This implies that one set out of $R_{2,m}^k$ and $L_{1,m}^k$ has the maximum number of intruders. In this scenario, the algorithm compares the two sets associated with each defender individually. In particular, the algorithm compares the number of intruders in the sets $L_{1,m}^k$ (resp. $R_{2,m}^k$) and $I_m^k$ for defender $V_1^m$ (resp. $V_2^m$) and then assigns the set with more number of intruders to defender $V_1^m$ (resp. $V_2^m$). If set $L_{1,m}^k$ (resp. $R_{2,m}^k$) has more number of intruders as compared to the set $I_m^k$, then defender $V_1^m$ (resp. $V_2^m$) moves counter clockwise (resp. clockwise). Otherwise, defender $V_1^m$ (resp. $V_2^m$) moves clockwise (resp. counter clockwise). This is mathematically described as follows.

**Algorithm 7:** Cooperative Compare-and-Capture (Coop-CaC) Algorithm

Select $x_C^{1,m} = x_C^{2,m} \in [\rho - r, \min\{\rho + r, \frac{3M(1-r)}{3M+4\theta v}\}]\}$.
**for** *each epoch* $k \geq 1$ **do**
  **for** *all* $s \in \{1, 2\}$ *and every dominance region* $m \in \{1, \ldots, \frac{M}{2}\}$ **do**
    Determine $\underline{\beta}_{s,m}, \overline{\beta}_{s,m}$ depending on Case 1 or Case 2 from the description.
  **end**
  Move every defender $V_s^m$ on $\mathcal{T}_s(x_C^{s,m}, \underline{\beta}_{s,m}, \overline{\beta}_{s,m})$.
  Move every defender $V_s^m$ back to starting position.
**end**

- For defender $V_1^m$

$$\underline{\beta}_{1,m} = \phi_1 - \tfrac{4\theta}{3M}, \overline{\beta}_{1,m} = \phi_1 \text{ if } |I_m^k| < |L_{1,m}^k|$$
$$\underline{\beta}_{1,m} = \phi_1, \overline{\beta}_{1,m} = \phi_1 + \tfrac{4\theta}{3M}, \text{ otherwise.}$$

- For defender $V_2^m$

$$\underline{\beta}_{2,m} = \phi_1, \overline{\beta}_{2,m} = \phi_1 + \tfrac{4\theta}{3M} \text{ if } |I_m^k| < |R_{2,m}^k|$$
$$\underline{\beta}_{2,m} = \phi_1 - \tfrac{4\theta}{3M}, \overline{\beta}_{2,m} = \phi_1, \text{ otherwise.}$$

Once the parameters of the angular paths of the defenders are set, the defenders move on the angular paths and then return to their starting locations. The next epoch begins once the defenders reaches their starting locations.

For the initial case, we assume time 0 as the time when the first intruder is released. Each defender $V_{s,m}$, $1 \leq s \leq 2$ in dominance region $m, m \in \{1, \ldots, \frac{M}{2}\}$ starts at location $(x_C^{s,m}, \phi_1)$ and begins its first epoch at time 0.

The following result characterizes the parameter regime for Algorithm Coop-CaC.

**Lemma 44.** *Any intruder in dominance region $m$ at a radial location exceeding $x_C^{s,m} + r + (\frac{8\theta}{3M} - \phi)vx_C^{s,m}$, $\forall \phi \in [0, \frac{4\theta}{3M}]$ and for any $s \in \{1, 2\}$, at the start of epoch $k$ is contained either in the set $R_{s,m}^{k+1}$ or in $L_{s,m}^{k+1}$ in epoch $k+1$ and is not lost at the start of epoch $k+1$ if*

$$v \leq \frac{3M(x_C^{s,m} + r - \rho)}{8\theta x_C^{s,m}}.$$

*Proof.* Without loss of generality, let $s = 2$ and assume that $|I_m^k| < |R_{2,m}^k|$ holds in dominance region $m \in \{1, \dots, \frac{M}{2}\}$. The total time taken by defender $V_2^m$ to capture intruders in $R_{2,m}^k$ and return back to its starting location $(x_C^{2,m}, \phi_1)$ is $\frac{8\theta}{3M} x_C^{2,m}$. In the worst-case, in order for any intruder $i$ to be not considered in the start of epoch $k$, the intruder $i$ must be located just above $(x_C^{2,m} + r + \frac{4\theta}{3M} v x_C^{2,m}, \phi_1 + \frac{4\theta}{3M})$ at the start of epoch $k$. By the time the defender reaches location $(x_C^{2,m}, \phi_1 + \frac{4\theta}{3M})$, intruder $i$ will be located just above the location $(x_C^{2,m} + r, \phi_1 + \frac{4\theta}{3M})$ and will not be captured.

If $v \leq \frac{3M(x_C^{s,m} + r - \rho)}{8\theta x_C^{s,m}}$ holds for dominance region $m$, intruder $i$ will be at least $\frac{4\theta}{3M} v x_C^{2,m}$ distance away from the perimeter at the end of epoch $k$. Thus, intruder $i$ gets considered at the start of epoch $k+1$ as per the definitions of the sets $L_{2,m}^{k+1}$ and $R_{2,m}^{k+1}$ implying that intruder $i$ is not lost at the start of epoch $k+1$. This concludes the proof. □

**Corollary 7.** *Any intruder in dominance region $m$ which lies beyond the set $I_m^k$ in epoch $k$ will be contained in the set $I_m^{k+1}$ in epoch $k+1$ if Lemma 44 holds.*

*Proof.* The proof directly follows from the fact that Lemma 44 holds for both defenders $V_1^m$ and $V_2^m$ in dominance region $m$. □

**Theorem 45.** *The competitive ratio $c_{Coop-CaC} = 1.5$ for any problem instance $\mathcal{P}$ that satisfies*

$$v \leq \min_{m \in \{1, \dots, M\}} \left\{ \frac{3M(x_C^{s,m} + r - \rho)}{8\theta x_C^{s,m}} \right\}$$

*and equation* (4.6) *for any* $x_C^{s,m} \in [\rho - r, \min\{\rho + r, \frac{3M(1-r)}{3M+4\theta v}\}]$.

*Proof.* The idea is to prove the result for any dominance region $m \in \{1, \dots, \frac{M}{2}\}$. The result then will follow from the fact that the motion of defenders in a dominance region is independent of the motion of the defenders in other dominance regions.

Let there be $r_2, l_1$ and $l_2 + r_1$ intruders in the sets $R_{2,m}^k$, $L_{1,m}^k$ and $I_m^k$, respectively, in a dominance region $m$. Then based on the number of intruders in the sets $R_{2,m}^k, L_{1,m}^k$ and $I_m^k$ in an epoch $k$, three cases arise.

**Case 1:** $|I_m^k| \geq |L_{1,m}^k|$ and $|I_m^k| \geq |R_{2,m}^k|$. This implies that the following two conditions must hold in epoch $k$.

$$r_1 + l_2 \geq l_1 \text{ and } r_1 + l_2 \geq r_2.$$

Further, without loss of generality, assume that $l_1 \geq r_2$ holds, i.e., the condition $|L_{1,m}^k| \geq |R_{2,m}^k|$ holds. Then, it follows that the two defenders jointly captured $r_1 + l_2 + l_1$ intruders out of $r_1 + l_2 + l_1 + r_2$ in epoch $k$. Assuming that an optimal offline algorithm can capture all intruders in the worst-case and using the aforementioned conditions, we obtain

$$c_{Coop-CaC} = \frac{r_1 + l_2 + l_1 + r_2}{r_1 + l_2 + l_1} \leq \frac{3r_2}{2r_2} = 1.5.$$

**Case 2:** $|I_m^k| < |L_{1,m}^k|$ and $|I_m^k| \geq R_{2,m}^k$ holds. This implies that the following two conditions must hold in epoch $k$.

$$r_1 + l_2 < l_1 \text{ and } r_1 + l_2 \geq r_2.$$

In this case, the defenders jointly capture $r_1+l_2+l_1$ intruders out of $r_1+l_2+l_1+r_2$ intruders. Thus, assuming that the optimal offline algorithm can capture all intruders and using the aforementioned conditions yields

$$c_{Coop-CaC} = \frac{r_1 + l_2 + l_1 + r_2}{r_1 + l_2 + l_1} \leq \frac{3r_2}{2r_2} = 1.5.$$

The proof for the case when $|I_m^k| \geq |L_{1,m}^k|$ and $|I_m| < R_{2,m}^k$ holds is omitted for brevity as it is analogous to Case 2.

**Case 3:** $|L_{1,m}^k| > |I_m^k|$ and $|R_{2,m}| > |I_m^k|$. This implies that the following two conditions must hold in epoch $k$.

$$r_1 + l_2 < l_1 \text{ and } r_1 + l_2 < r_2.$$

In this case, the defenders jointly capture $l_1 + r_2$ intruders out of $r_1 + l_2 + l_1 + r_2$ intruders. Following similar steps as in previous two cases yields

$$c_{Coop-CaC} = \frac{r_1 + l_2 + l_1 + r_2}{r_2 + l_1} \leq \frac{3(r_1 + l_2)}{2(r_1 + l_2)} = 1.5.$$

Thus, in any epoch $k$ and in a dominance region $m \in \{1, \ldots, \frac{M}{2}\}$, the two defenders contained in dominance region $m$ jointly capture at least $\frac{2}{3}^{rd}$ of the total intruders that arrive in that particular dominance region. Since, the number of intruders captured by the defenders in any dominance region $m \in \{1, \ldots, \frac{M}{2}\}$ does not depend on any other dominance regions, it follows that in every dominance region, the defenders capture at least $\frac{2}{3}^{rd}$ of the total intruders that arrive in their respective dominance region. This implies that Algorithm Coop-CaC captures at least $\frac{2}{3}^{rd}$ of the total intruders that arrive in the environment. Finally, Lemma 44 and Corollary 7 ensures that every intruder is considered in the sets $L^k_{1,m}, R^k_{2,m}$ and $I^k_m$. Since every intruder is considered and the defenders capture at least $\frac{2}{3}$-rd of the total number of intruders it follows that $c_{Coop-CaC} = 1.5$ and the result is established. $\square$

**Remark 4.** *As* $\frac{3M(x^{s,m}_C + r - \rho)}{8\theta x^{s,m}_C}$ *is a monotonically increasing function of* $x^{s,m}_C$*, positioning each defender* $V^m_s, 1 \leq s \leq 2$ *in every dominance region* $m \in \{1, \ldots, \frac{M}{2}\}$ *at* $\min\{\rho + r, \frac{3M(1-r)}{3M+4v\theta}\}$ *yields* $v \leq \min\left\{\frac{3Mr}{4\theta(\rho+r)}, \frac{3M(1-\rho)}{4\theta(2-3r+\rho)}, \frac{3M(1-\rho)}{4\theta(\rho-r)}\right\}$.

**Remark 5.** *The maximum speed ratio for which Algorithm Coop-CaC is effective is* $3/4$*-th of the maximum speed ratio for which Algorithm Dec-CaC is effective.*

Recall that in Coop-CaC algorithm, the motion of the defenders in a dominance region is independent of the motion of the defenders contained in another dominance region. Due to this, Coop-CaC Algorithm requires an even number of defenders. Next, we propose a variation of Algorithm Coop-CaC that relaxes this requirement, i.e., the next algorithm is effective for any value of $M > 1$. An important distinction to note between the two algorithms is that in Algorithm Coop-CaC, every defender shares at most one sector whereas in the Modified Coop-CaC algorithm, every defender shares at least one and at most two sectors. As Modified Coop-CaC algorithm is similar to Coop-CaC algorithm, we briefly describe the algorithm and its competitive ratio in the following remark and omit establishing the result formally due to space constraints.

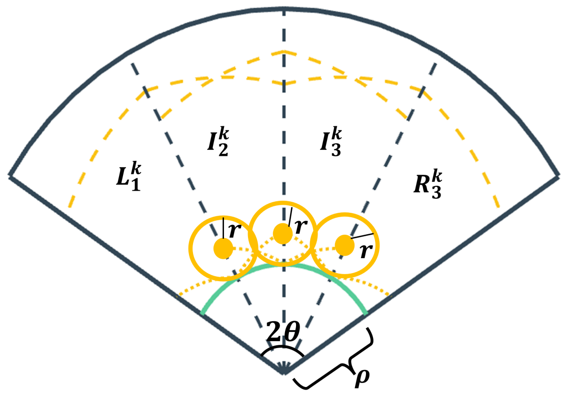



Figure 4.11 Description of Modified Coop-CaC Algorithm for an epoch $k$ and $M = 3$. Set $I_2^k$ (resp. $I_3^k$) denote the intersection of the sets $R_1^k$ and $L_2^k$ (resp. $R_2^k$ and $L_3^k$).

**Remark 6** (Modified Coop-CaC)**.** *Modified Coop-CaC algorithm partitions the environment into $M+1$ sectors, each of angle $\frac{2\theta}{M+1}$ and positions the defenders such that the sets $R_m^k$ and $L_{m+1}^k$ overlap, for all $m \in \{1, \ldots, M-1\}$ (cf. Fig. 4.11). Then, following similar steps as for Algorithm Coop-CaC, it can be shown that the Modified Coop-CaC algorithm is $\frac{M+1}{M}$-competitive. Further, it can be shown by following the steps of Algorithm Coop-CaC that the parameter regime of the modified algorithm is lower than that of the Coop-CaC algorithm.*

Although the parameter regime of Modified Coop-CaC is lower than that of Algorithm Coop-CaC, note that $c_{Modified-CaC} < c_{Coop-CaC}$ for any $M > 2$ and $c_{Modified-CaC} = c_{Coop-CaC}$ for $M = 2$. We now present our second cooperative algorithm.

#### 4.3.2.2 Cooperative Stay Near Perimeter (Coop-SNP) Algorithm

The idea behind this algorithm is to divide the entire environment $\mathcal{E}(\theta)$ into $N = \lceil \frac{\theta}{\theta_s} \rceil$ sectors, each of angle $2\theta_s = 2\arctan(\frac{r}{\rho})$. We highlight an important distinction between Algorithm Coop-SNP and Algorithm Dec-SNP. In Algorithm Dec-SNP, the environment is first partitioned into $M$ dominance regions and then each dominance region is further divided into $n_s$ sectors. On the other hand, in Algorithm Coop-SNP, we directly divide the entire environment into $N$ sectors. We assume that the number of defenders is strictly less than the number of sectors, i.e., $M < N$. If this does not hold, then by assigning a defender to every sector, Coop-SNP will capture all intruders, yielding the competitive ratio as 1. Let

**Algorithm 8:** Cooperative Stay Near Perimeter Algorithm

Assumes each defender $V_m, 1 \leq m \leq M$ is at $(x_m, \alpha_m)$ in sector $N_m$
**for** *each* $j \geq 1$ **do**
    Determine sets $\mathbf{O}, \hat{\mathbf{O}}$ and $\tilde{\mathbf{O}}$.
    **for** *every sector* $\delta_g \in \tilde{\boldsymbol{O}}$ **do**
        Assign the defender located in $\delta_g$ to sector $\delta_g$ for $2\tilde{D}$ time units.
    **end**
    **for** *every sector* $\delta_q \in \hat{\boldsymbol{O}}$ **do**
        **if** $|S_q^{j+2}| \geq |S_p^{j+1}|$ **then**
            Assign $V_p$ to sector $\delta_q$ for $2\tilde{D}$ time units.
            Remove $V_p$ from set $\hat{\mathbf{V}}$ and add to set $\mathbf{V}_2$.
        **else**
            Assign $\delta_p$ for $\tilde{D}$ time units
            Assign $\delta_q$ for $\tilde{D}$ time units.
            Remove $V_p$ from set $\hat{\mathbf{V}}$ add to set $\mathbf{V}_1$.
        **end**
    **end**
    Defenders in $\tilde{\mathbf{V}}$ stay at their location for $2\tilde{D}$ time.
    **for** *every* $V_q \in \boldsymbol{V}_2$ **do**
        Move to the resting point of assigned sector.
        Stay at the resting point until time $2(j+2)\tilde{D}$.
    **end**
**end**
**for** *every* $V_q \in \boldsymbol{V}_1$ **do**
    Stay location until time $2(j+1)\tilde{D}$.
    Move to resting point of the second sector assigned.
**end**

$\delta_l, l \in \{1, \ldots, N\}$, denote the $l^{th}$ sector, where $\delta_1$ denotes the leftmost sector and we assume that the sectors are numbered from left to right. Further, let $\partial R_l$ denote the portion of the perimeter contained in sector $\delta_l$. Then, a resting point $(x_l, \alpha_l) \in \mathcal{E}(\theta)$ of a sector $\delta_l$ is defined as a location for defender $V_m$ such that when positioned at that location, the entire $\partial R_l$ is contained completely within the capture radius of the defender $V_m$. Mathematically,

$$(x_l, \alpha_l) := \Big(\rho \cos(\theta_s), \Big(l - \frac{N+1}{2}\Big)2\theta_s\Big).$$

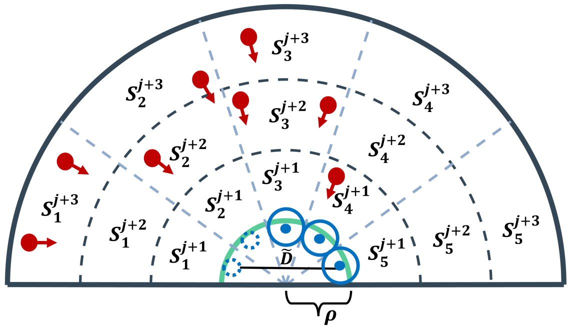



Figure 4.12 Description of Coop-SNP Algorithm. Blue dashed circles denote the resting points. Defender $V_1, V_2$ and $V_3$ are located in $\delta_3, \delta_4$, and $\delta_5$, respectively. $\mathbf{O} = \{\delta_1, \delta_2, \delta_3\}$, $\tilde{\mathbf{O}} = \{\delta_3\}$, $\hat{\mathbf{O}} = \{\delta_2, \delta_1\}$, $\tilde{\mathbf{V}} = \{V_1\}$ and $\hat{\mathbf{V}} = \{V_3, V_2\}$.

Lastly, let $\tilde{D}$ denote the distance between the two resting points that are furthest apart (cf. Fig. 4.12). Formally,

$$\tilde{D} = \begin{cases} 2\rho\cos(\theta_s)\sin((N-1)\theta_s), \text{ if } (N-1)\theta_s < \frac{\pi}{2} \\ 2\rho\cos(\theta_s), \text{ otherwise.} \end{cases} \tag{4.7}$$

After dividing the environment into $N$ sectors, analogous to Dec-SNP algorithm, Coop-SNP algorithm radially divides the environment $\mathcal{E}(\theta)$ into three intervals of length $\tilde{D}v$ or equivalently, time intervals of time length $\tilde{D}$ each. Specifically, the $j^{th}$ time interval for any $j > 0$ is defined as the time interval $[(j-1)\tilde{D}, j\tilde{D}]$. The requirement to ensure a finite competitiveness for Coop-SNP algorithm is the same is Dec-SNP, i.e., we require $\frac{1-\rho}{v} \geq 3\tilde{D}$. For any $j \geq 1$, let $S_l^j$ be the set of intruders that arrive in a sector $\delta_l$ in the $j^{th}$ time interval.

The Coop-SNP algorithm, defined in Algorithm 8, is based on the following two steps: 1) For every $j \geq 0$, at time instant $3\tilde{D} + 2j\tilde{D}$, Coop-SNP selects $M$ sectors with highest number of intruders and, 2) assigns $M$ defenders to the resting locations of the $M$ sectors selected in the first step. We describe each step in more detail below.

**Comparison (S1)**: Coop-SNP determines $M$ sectors which has the most number of intruders. In particular, suppose that defenders $V_1, \ldots, V_M$ are located at the resting points of sectors $\delta_1, \ldots, \delta_M$, respectively, at the $j$th iteration. Let $\mathbf{L}$ denote the set of those corresponding indices. Then, for every $i \in \{1, \ldots, N\}$, let

$$\eta_{\mathbf{L}}^i = \begin{cases} |S_i^{j+2}| + |S_i^{j+3}|, & \text{if } i \notin \mathbf{L}, \\ |S_i^{j+1}| + |S_i^{j+2}| + |S_i^{j+3}|, & \text{otherwise.} \end{cases}$$

Coop-SNP then selects the top $M$ sectors which have the highest number of intruders. Specifically, Coop-SNP sorts the set $\{\eta_{\mathbf{L}}^1, \ldots, \eta_{\mathbf{L}}^N\}$ in a decreasing order and then selects the sectors corresponding to the first $M$ elements obtained from the ordered set. If there are multiple sectors with the same number of intruders, then Coop-SNP breaks the tie analogously to Dec-SNP algorithm.

We introduce the following notations for the second step. Let $\mathbf{O}$ be a set of sectors chosen as the outcome of comparison **S1** and let $\tilde{\mathbf{O}} \subseteq \mathbf{O}$ denote the set of those sectors which have defenders already positioned at their respective resting points (cf. Fig. 4.12). Let $\tilde{\mathbf{V}}$ denote the set of defenders that are located at the resting points of the sectors in $\tilde{\mathbf{O}}$. Note that $|\tilde{\mathbf{V}}| = |\tilde{\mathbf{O}}|$. For the sectors in $\hat{\mathbf{O}} = \mathbf{O} \setminus \tilde{\mathbf{O}}$, we assume that the sectors in the set $\hat{\mathbf{O}}$ are ordered with respect to the number of intruders in their second, i.e., $(j+2)^{th}$, interval (cf. Fig. 4.12). Specifically, the first (resp. last) element of $\hat{\mathbf{O}}$) has the most (resp. least) number of intruders in the $(j+2)^{th}$ interval. Similarly, let $\hat{\mathbf{V}} = \mathbf{V} \setminus \tilde{\mathbf{V}}$ be the set of defenders not located in the sectors contained in $\tilde{\mathbf{O}}$. We assume that the set $\hat{\mathbf{V}}$ is in an increasing order according to the number of intruders in the first, i.e., $(j+1)^{th}$ interval of the sectors the defenders are located in (cf. Fig. 4.12).

**Assignment (S2)**: The aim of this step is to ensure that every sector in the set $\mathbf{O}$ is assigned a defender. For every element $\delta_q \in \mathbf{O}$, Algorithm Coop-SNP does the following:

i For all $\delta_q \in \tilde{\mathbf{O}}$, Coop-SNP assigns the defender that is already located in $\delta_q$ to sector $\delta_q$ for $2\tilde{D}$ time units.

ii For every sector $\delta_q \in \hat{\mathbf{O}}$,

a) Let $V_p$ be the first element of the set $\hat{\mathbf{V}}$ and suppose that $V_p$ is located in sector $\delta_p$. Then Coop-SNP assigns defender $V_p$ to sector $\delta_q \in \hat{\mathbf{O}}$ for $2\tilde{D}$ time units if

$|S_q^{j+2}| > |S_p^{j+1}|$ holds. After assigning $V_p$ to $\delta_q$, Coop-SNP removes $V_p$ from the set $\hat{\mathbf{V}}$.

b) Otherwise, Coop-SNP first assigns the sector $\delta_p$ to defender $V_p$ for $\tilde{D}$ time units and then sector $\delta_q$ for the next $\tilde{D}$ time units. Coop-SNP then removes $V_p$ from the set $\hat{\mathbf{V}}$.

Once the assignment **S2** is complete, each defender in the set $\tilde{\mathbf{V}}$ captures the first two intervals in their respective sectors by staying at their locations for $2\tilde{D}$ time units. The defenders that are assigned sectors based on point (ii-a) of **S2**, move to the resting points of their assigned sectors, reaching the location in at most $\tilde{D}$ time units, and then stay at that location until all of the intruders in the $(j+2)^{th}$ interval are captured. Finally, the defenders that are assigned sectors based on point (ii-b) of **S2**, first stay at their locations until all of the intruders in the $(j+1)^{th}$ interval are captured. Then, they move to the resting point of the assigned sectors.

At time 0, each defender $V_i, i \in \{1, \ldots, M\}$ is located at the resting point of sector $\delta_i$ and waits at its location until $\tilde{D}$ time units. Then, setting $|S_i^{j+1}| = |S_i^{j+2}| = 0$, the steps **S1** and **S2** are applied. The first iteration then begins at time $3\tilde{D}$.

**Lemma 46.** *For any $j \geq 1$, every defender $V_m \in \mathbf{V}$, assumed to be located in sector $\delta_m$ and assigned sector $\delta_i$, captures at least one intruder interval, i.e., either $S_m^{j+1}$ or $S_i^{j+2}$, in the time interval $[j\tilde{D}, (j+2)\tilde{D}]$.*

*Proof.* Algorithm Coop-SNP assigns a sector to a defender in only three possible ways. First, if $\delta_m = \delta_i$ (point (i) of **S2**), then the result follows as the defender stays at its current location and captures interval $S_m^{j+1}$ and $S_m^{j+2}$ of sector $\delta_i$. Second, sector $\delta_i$ is assigned for $2\tilde{D}$ time (point (ii-a) of **S2**), then the defender moves to sector $\delta_i$ and then captures $S_i^{j+2}$. Finally, if sector $\delta_m$ is assigned followed by sector $\delta_i$ (point (ii-b) of **S2**), then the defender stays at its location for $\tilde{D}$ time and captures $S_m^{j+1}$. Thus, in all three cases, the defender captures at least one interval. Since, this proof holds for every defender in $\mathbf{V}$, the result is established. □

Similar to Dec-SNP algorithm, to establish the competitive ratio of Algorithm Coop-SNP, we use an accounting analysis in which captured intervals *pay* for the lost intervals or equivalently, captured intervals are *charged* for the intervals lost. A *charge* to the captured interval(s) consists of all the lost intervals that are considered in a single comparison step **S1** involving that particular captured interval(s).

The following two lemmas will jointly establish the competitive ratio of Coop-SNP algorithm, proofs of which are provided in the Appendix.

**Lemma 47.** *Under Algorithm Coop-SNP, for each defender, every two consecutive captured intervals are charged at most* $3(N-M)$ *times.*

*Proof.* Please see Appendix 12. □

We now establish that each lost interval is fully accounted for by the captured intervals.

**Lemma 48.** *Every lost interval is accounted for by the captured intervals of Algorithm Coop-SNP.*

*Proof.* Please see Appendix 12. □

**Theorem 49.** *Algorithm Coop-SNP is* $\frac{3N-M}{2M}$*-competitive for any problem instance* $\mathcal{P}$ *which satisfies* $3\tilde{D} \leq \frac{1-\rho}{v}$*, where* $N = \lceil \frac{\theta}{\theta_s} \rceil$*,* $\theta_s = \arcsin(\frac{r}{\rho})$ *and* $\tilde{D}$ *is defined in equation* (4.7)*.*

*Proof.* From Lemma 47, every two consecutive captured intervals for each defender account for $3(N-M)$ lost intervals implying that every two captured intruders pay for at most $3(N-M)$ intruders. Since there are total of $M$ defenders, the total number of intervals are $3(N-M)+2M = 3N-M$. Lemma 48 ensures that every lost interval is fully accounted by the captured intervals. Thus, assuming that the optimal offline captures all intruder intervals, we obtain the result. □

**Corollary 8.** *Given a problem instance* $\mathcal{P}$*,* $c_{Coop\text{-}SNP} \leq c_{Dec\text{-}SNP}$*.*

*Proof.* Recall from Theorem 43 and Theorem 49, the competitive ratio of Algorithm Dec-SNP and Algorithm Coop-SNP is $\frac{3n_s-1}{2}$ and $\frac{3N-M}{2M}$, respectively. The idea is to show that for any problem instance $\mathcal{P}$, $\frac{3n_s-1}{2} \geq \frac{3N-M}{2M}$ or equivalently, $n_s \geq \frac{N}{M}$.

Since $N > M$ is an integer, let $\lceil \frac{\theta}{\theta_s} \rceil = M + p$, where $p \geq 1 \in \mathbb{Z}^+$. This yields

$$\frac{N}{M} = \left(1 + \frac{p}{M}\right).$$

Two cases arise:

**Case 1:** If $\frac{\theta}{\theta_s} = M + p$, i.e., $\frac{\theta}{\theta_s}$ is an integer. Then

$$n_s = \lceil 1 + \frac{p}{M} \rceil \geq 1 + \frac{p}{M},$$

and we obtain the result.

**Case 2**: If $\frac{\theta}{\theta_s}$ is not an integer. Then, for an $0 < \epsilon < 1$,

$$n_s = \lceil 1 + \frac{p-1+\epsilon}{M} \rceil.$$

If $p \leq M$, it follows that $\lceil 1 + \frac{p-1+\epsilon}{M} \rceil \geq 1 + \frac{p}{M}$ and the result holds. For any $p > M$, the result follows as $\frac{p}{M}$ can be expressed as $q + \frac{l}{M}$, where $q \in \mathbb{Z}^+$ and $0 < l < M$ yielding a similar expression as in Case 2. This concludes the proof. □

**Remark 7** (Practical Considerations)**.** *While Algorithm Coop-SNP can be applied as is to UAVs flying at different altitudes, collision avoidance algorithms may be required to be implemented in other cases when the vehicles move from their respective sectors to some other sectors. In such cases, the parameter $\tilde{D}$ must be suitably scaled to account for the extra time required by the vehicle to move to a sector. Doing so may lower the upper bound on parameter $v$ characterized for the algorithm but the bound on the competitive ratio still holds as long as $3\tilde{D} \leq (1-\rho)/v$ holds, as the competitive ratio of Algorithm Coop-SNP does not depend on the value of $\tilde{D}$.*

**Remark 8.** *Although the competitive ratio of Coop-SNP as well as Coop-CaC algorithm is lower than that of Dec-SNP and Dec-CaC, respectively, the cooperative algorithms require*

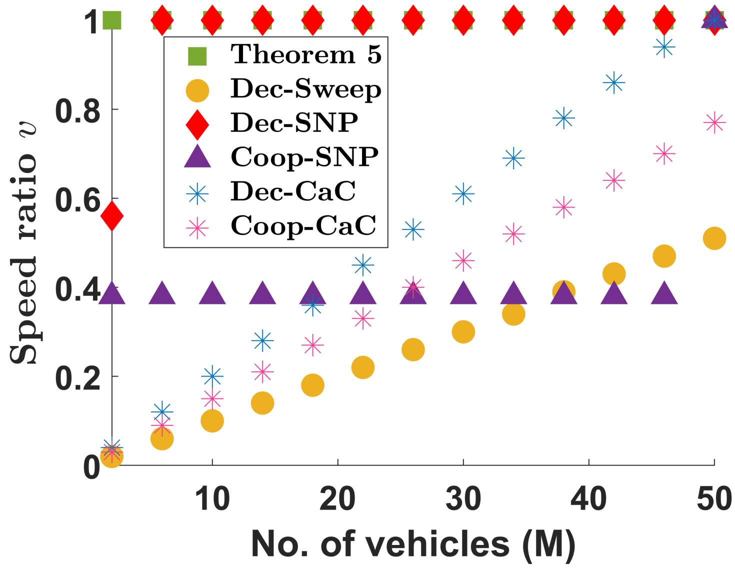



Figure 4.13 $(v, M)$ parameter regime plot for fixed $\rho = 0.3, r = 0.01$, and $\theta = \frac{\pi}{2}$. For these values of parameters, a $c$-competitive algorithm may exist for all values of $v \leq 1$ (Theorem 35). The algorithms designed in this work have a finite competitive ratio for values of $v$ that lie below their corresponding markers. In particular, Algorithm Dec-Sweep, Dec-CaC and Coop-CaC are 1, 2 and 1.5-competitive, respectively, for all values of $M$ and for values of $v$ below their corresponding markers. The competitive ratio of both Algorithm Dec-SNP and Coop-SNP varies with the parameter $M$. The parameter regime of Algorithm Coop-SNP does not depend on $M$.

*intruders to be slower than the decentralized algorithms. Equivalently, the cooperative algorithms require higher number of vehicles to cover the same $(v, \rho)$ parameter regime as the decentralized algorithms but achieve a lower competitive ratio.*

## 4.4 Numerical Observations and Discussions

We now provide a numerical visualization of the analytical bounds derived in this chapter. We focus on the parameter regimes $(v, M)$ and $(v, \rho)$. We also compare the competitive ratios of the decentralized and the centralized SNP algorithms and provide insights into the case of heterogeneous capture radii and perimeters of arbitrary shapes.

### 4.4.1 $(v, M)$ parameter regime

Figure 4.13 and Figure 4.14 shows the $(v, M)$ parameter regime plot for the fixed value of $\rho, r$, and $\theta$.

Observe that as the number $M$ of defenders increases, the curves for Algorithms Dec-

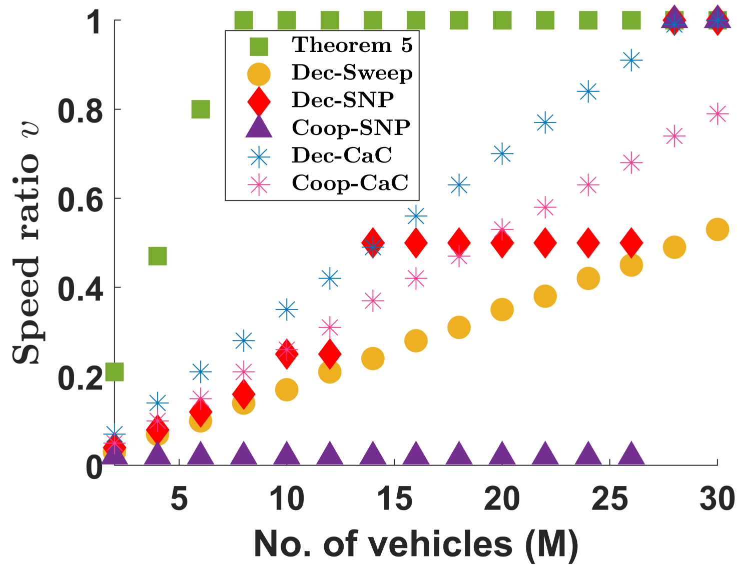



Figure 4.14 $(v, M)$ parameter regime plot for fixed $\rho = 0.85, r = 0.05$, and $\theta = \frac{\pi}{2}$. For any value of $v$ that lies beyond the green markers, there does not exist a $c$-competitive algorithm. The algorithms designed in this work have a finite competitive ratio for values of $v$ that lie below their corresponding markers. Specifically, Algorithm Dec-Sweep, Dec-CaC and Coop-CaC are 1, 2 and 1.5-competitive, respectively, for all values of $M$ and values of $v$ that lie below their respective markers. The competitive ratio of both Algorithm Dec-SNP and Coop-SNP varies with the parameter $M$. The parameter regime of Algorithm Coop-SNP does not depend on $M$.

Sweep, Dec-CaC, Coop-CaC and Dec-SNP approach the curve depicting the condition of Theorem 35. This implies that it is possible to capture faster intruders with a greater number of defenders. Further, while the curve for Algorithm Dec-Sweep, Algorithm Dec-CaC and Algorithm Coop-CaC are linear, the curve for Algorithm Dec-SNP is in *regions*, with each region having a specific competitive ratio. Since the condition in Theorem 49 does not depend on $M$, the curve for the conditions of Algorithm Coop-SNP is a straight line. Given the values of the parameters $\rho, r, \theta$ for Figure 4.13, note that the green curve is constant $v = 1$ for all values of $M$, implying that there may exist a $c-$competitive algorithm in this parameter regime. Interestingly, the curve for Dec-SNP algorithm reaches the green curve. Finally, note that the competitive ratio of Algorithm Coop-SNP is strictly less than 4 for $M > 16$ whereas the competitive ratio of Algorithm Dec-SNP is greater than 4 for all $2 \leq M \leq 20$.

In Figure 4.14, the curve for Algorithm Dec-SNP is never above the curve for Algorithm

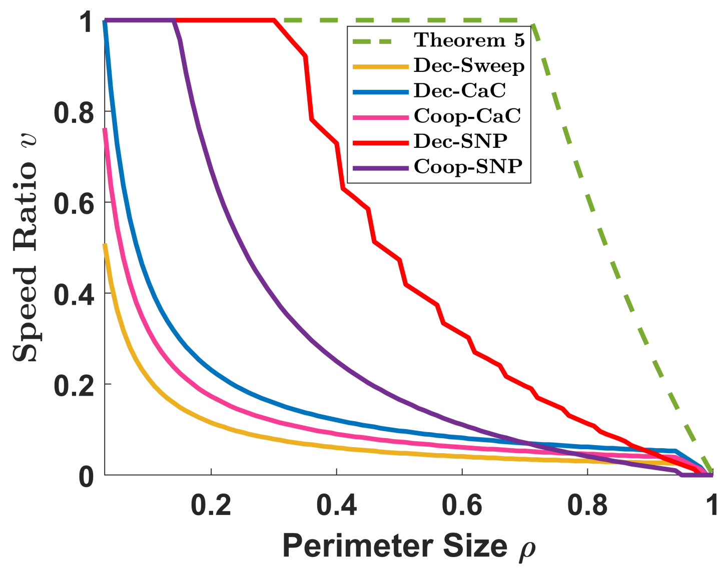



Figure 4.15 $(v, \rho)$ parameter regime plot for fixed $r = 0.02, \theta = \frac{\pi}{2}$ and $M = 4$. No $c$-competitive algorithm exists beyond the green curve. For values of $v$ below the yellow (resp. blue) curve, Algorithm Dec-Sweep (resp. Dec-CaC) is 1 (resp. 2)-competitive. For values of $v$ below the pink curve, Algorithm Coop-CaC is 1.5-competitive. Competitive ratio of Algorithms Dec-SNP and Coop-SNP varies with parameter $\rho$.

Dec-CaC implying that Dec-CaC is more effective for high values of $\rho$ unless Algorithm Dec-SNP has a competitive ratio less than 2. Specifically, for $M < 27$, Algorithm Dec-SNP has a competitive ratio strictly greater than 2 and its curve lies below that of Algorithm Dec-CaC and thus, Algorithm Dec-CaC is more effective. For $M > 27$, Algorithm Dec-SNP as well as Algorithm Coop-SNP are 1-competitive. Thus, for high values of $M$, Algorithm Dec-SNP is more effective. However, since the curve of Algorithm Coop-SNP lies below the curve of Dec-Sweep, it is not effective.

### 4.4.2 $(v, \rho)$ parameter regime

Figure 4.15 and Figure 4.16 shows the $(v, \rho)$ parameter regime for fixed values of $r, M$ and $\theta$.

From both Figure 4.15 and Figure 4.16, since the competitiveness of Algorithm Coop-SNP and Algorithm Dec-SNP is based on the number of sectors, the curve for both algorithms is divided into *regions*, where each region corresponds to a constant value of competitiveness. As the capture radius $r$ increases, the number of regions decreases. Similarly, the number

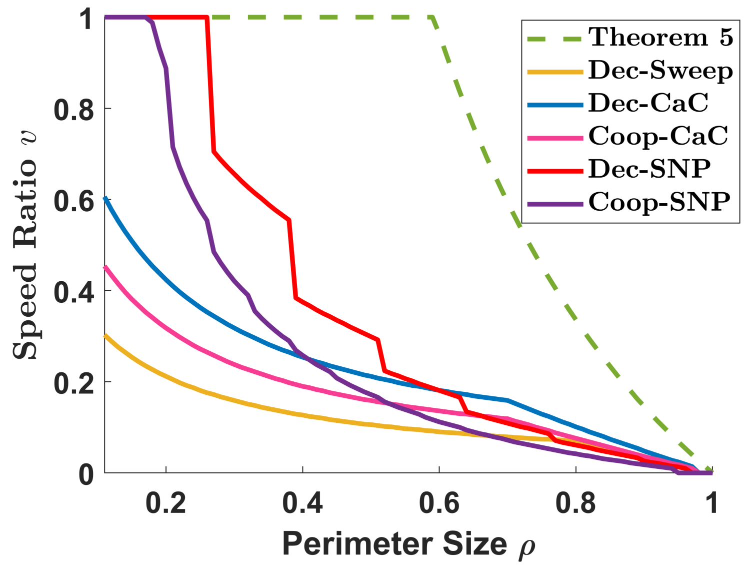



Figure 4.16 $(v, \rho)$ parameter regime plot for fixed $r = 0.1, \theta = \frac{\pi}{2}$ and $M = 2$. No $c$-competitive algorithm exists beyond the green curve. For values of $v$ below the yellow (resp. blue) curve, Algorithm Dec-Sweep (resp. Dec-CaC) is 1 (resp. 2)-competitive. For values of $v$ below the pink curve, Algorithm Coop-CaC is 1.5-competitive. Competitive ratio of Algorithms Dec-SNP and Coop-SNP varies with parameter $\rho$.

of regions increases as $\theta$ increases. Therefore, an important feature of Algorithm Coop-SNP and Algorithm Dec-SNP is that they can be used to determine the tradeoff between the competitiveness and the required parameter regime for the problem instance.

In Figure 4.15 (resp. Figure 4.16), Algorithm Dec-CaC is more effective than Algorithm Dec-SNP and Algorithm Coop-SNP as the curves for Algorithm Dec-SNP and Algorithm Coop-SNP are below the curve for Algorithm Dec-CaC for $\rho > 0.76$ (resp. $\rho > 0.38$) and $\rho > 0.88$ (resp. $\rho > 0.57$), respectively. A similar observation is made for Algorithm Coop-CaC, Algorithm Dec-SNP and Algorithm Coop-SNP. Further, compared to that in Figure 4.16, both Algorithm Dec-SNP and Algorithm Coop-SNP have a larger area of utility suggesting that with a smaller capture radius, Dec-SNP and Coop-SNP can capture faster intruders, and it covers a larger area in the parameter regime but at the cost of higher competitive ratio.

In Figure 4.16, Algorithm Dec-SNP is 2.5-competitive for $0.15 \leq \rho \leq 0.26$ whereas Algorithm Coop-SNP is 1.75-competitive for $\rho \leq 0.2$ and 2.5-competitive for $0.2 < \rho \leq 0.26$.

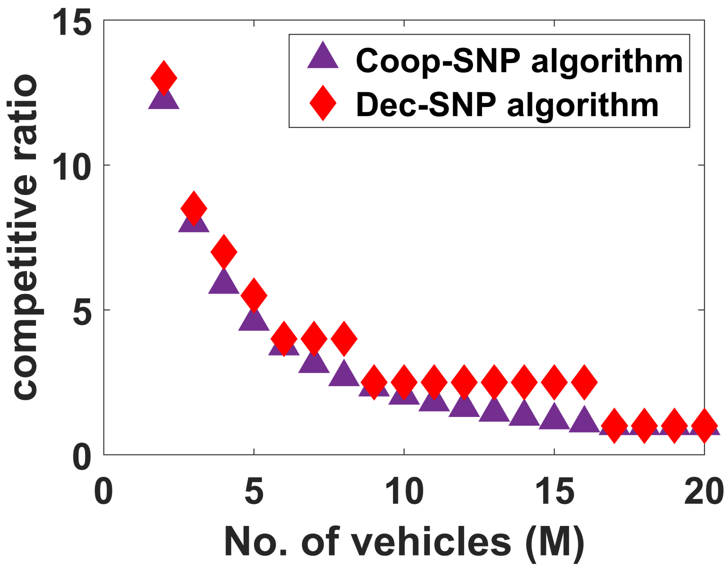



Figure 4.17 Comparison of competitive ratios for Algorithm Dec-SNP and Algorithm Coop-SNP for $\theta = \pi, r = 0.13$ and $\rho = 0.7$. Competitive ratio of Algorithm Coop-SNP is always lower than that of Algorithm Dec-SNP.

Both Dec-SNP and Coop-SNP are 1-competitive for $\rho < 0.15$. Similarly, in Figure 4.15, both Algorithm Dec-SNP and Algorithm Coop-SNP are 1-competitive for $\rho \leq 0.05$ suggesting that all intruders can be captured by increasing the number of vehicles.

### 4.4.3 Comparison of competitive ratios of Dec-SNP and Coop-SNP

Figure 4.17 shows the different values of competitive ratios for Algorithm Dec-SNP and Algorithm Coop-SNP for a fixed $\theta = \pi, r = 0.13$ and $\rho = 0.7$. Note that the competitive ratio of Algorithm Dec-SNP is constant for some values of $M$ whereas the competitive ratio of Algorithm Coop-SNP decreases as $M$ increases. For instance, Algorithm Dec-SNP is 2.5 competitive for $9 \leq M \leq 16$ whereas Algorithm Coop-SNP has a competitive ratio strictly less than 2.5 for the same values of $M$.

### 4.4.4 Extension to defenders with heterogeneous capture radii

We now describe how this work can be generalized to heterogeneous defenders having different capture radii $r_1, r_2, \ldots, r_M$.

Recall from the proof of Lemma 31, to determine the dominance region for the $M$ defenders, we use the fact that the time taken by the neighboring defenders must be the same to reach the endpoint of the common boundary. When the defenders have different capture radii, then following similar steps, one can obtain $M$ equations. Solving the $M$ equations will yield the angles for the $M$ dominance regions. If $2\beta_1, 2\beta_2, \ldots, 2\beta_M$ denote the angles of the $M$ dominance regions, then replacing $\frac{\theta}{M}$ in Lemma 31 and Lemma 32 by $\beta_i, 1 \leq i \leq M$ yields the position for defender $V_i$.

We now describe how Theorem 35 extends to defenders having different capture radii. The key idea behind the proof of Theorem 35 is to ensure that no defender can capture the burst intruders. From the partition angles $2\beta_i, \forall i \in \{1, \ldots, M\}$, one can compute the time taken by any defender $V_i$ to reach the endpoint of its dominance region. Since this time is same for all defenders $V_1, \ldots, V_M$, the time taken by any defender $V_i, i \in \{1, \ldots, M\}$, except $V_1$, to reach the endpoint of its dominance region is $\rho \sin(\beta_i) - r_i$. Here, without loss of generality, we assumed that defender $V_1$ captures an intruder from the stream of intruders and thus, the time taken by $V_1$ is $2\rho \sin(\beta_1) - 2r_1$ and thus, the result is obtained.

Finally, once the dominance region for $M$ defenders is obtained, then all of our decentralized algorithms can be directly applied. Their respective analytical lower bounds can also be obtained analogously. Our conjecture is that even when the defenders have different capture radii, Algorithm Dec-Sweep and Algorithm Dec-CaC are still 1 and 2-competitive, respectively.

We now briefly comment on the choice of algorithms and the applicability of this work. The choice of which algorithm to use depends on the problem parameters, i.e., the size of the perimeter and the speed ratio, and the acceptable bound on competitiveness. For instance, given the speed of the intruders and an acceptable bound on the competitive ratio, one may select a larger or smaller perimeter size. Note that it may not always be feasible to select a smaller perimeter size. In that case, faster defenders or algorithms with higher value of competitive ratio may be chosen. Further, the acceptable competitive ratio is governed by the application. For instance, defense applications may need $c = 1$ while agricultural applications may tolerate higher values of $c$. Finally, we remark that the sufficient conditions on the algorithms are derived in the worst-case. In some applications, these worst-case inputs may not be realizable and thus, the algorithms will perform better for those applications.

We now briefly describe how this work extends to perimeters with arbitrary shapes. Recall that the two important quantities that are used in the analysis are the time taken by the intruders to reach the perimeter and the time taken by a vehicle to move from one point to the other, while capturing intruders. For successful capture of an intruder, the time taken by the vehicle must be at most the time taken by the intruder. This work extends to perimeters with arbitrary shapes as these two quantities can be determined for any given (closed) shape of a perimeter. If the shape of the perimeter is such that the dominance region for each vehicle ensures the same time to move from one point to another for every vehicle, then the upper bound on $v$ is characterized by the maximum time taken by a vehicle to move within its dominance region (for instance angular motion in Dec-Sweep or Coop-CaC algorithm). Otherwise, the upper bound on $v$ is characterized by the vehicle in a particular dominance region which requires the maximum time for a vehicle. This is because in the worst-case, all of the intruders can be released in that vehicle's dominance region.

Through this work, we aim to study how an adversary can overwhelm the defense by using many (possibly infinite) attackers. This is useful to obtain insights about the problem including different adversarial inputs such as the one designed in the proof of Theorem 35. The insights gained on the worst-case inputs can be used for perimeter defense problems with many *responsive* intruders, i.e., intruders that can move in any direction. Note that the some of our results, including the fundamental limit and the Decentralized Sweep algorithm applies as is when responsive intruders are considered.

## 4.5 Conclusions

This work analyzed the problem wherein $M$ defenders, each having a finite capture radius, are tasked to defend a perimeter in a conical environment from intruders that are released arbitrarily in the environment. Our approach was based on competitive analysis that first yielded a fundamental limit on the problem parameters for finite competitiveness of any online algorithm. We then designed and analyzed five algorithms, i.e., Decentralized Sweep, Decentralized CaC, Decentralized SNP, Cooperative CaC and Cooperative SNP and established sufficient conditions that guaranteed a finite competitive ratio for each algorithm. In particular, we establish that Decentralized Sweep, Cooperative CaC and Decentralized CaC are 1, 1.5, and 2-competitive, respectively. Further, their competitive ratio increases linearly with $M$. The competitive ratio of Decentralized SNP and Cooperative SNP changes with the function of the parameters and does not always extend beyond that of Decentralized CaC algorithm in specific parameter regimes.

# CHAPTER 5

# COMPETITIVE PERIMETER DEFENSE WITH A TURRET AND A MOBILE DEFENDER

Until now, we focused on perimeter defense problems with homogeneous defenders. However, there are many scenarios when there are different types of defenders, i.e., heterogeneous defenders that must cooperate with each other to capture intruders. In this chapter, contents of which are based on [5], we look at two such heterogeneous defenders.

This chapter addresses a perimeter defense problem in a conical environment. The environment contains two heterogeneous defenders, namely a turret and a mobile vehicle, which seek to defend a perimeter by capturing mobile intruders. The intruders are released at the boundary of the environment and move radially inwards with fixed speed toward the perimeter. Defenders have access to intruder locations only after they are released in the environment. Further, the defenders have distinct motion and capture capability and thus, are heterogeneous in nature. Specifically, the vehicle, having a finite capture radius, moves with unit speed in the environment whereas the turret has a finite range and can only turn clockwise or anti-clockwise with a fixed angular rate. Jointly, the defenders aim to capture as many intruders as possible before they reach the perimeter. This is an *online* problem as the input, which consists of the total number of intruders, their release locations, as well as their release times, is gradually revealed over time to the defenders. Thus, we focus on the design and analysis of online algorithms to route the defenders. Aside from military applications, this work is also motivated by monitoring applications wherein a drone and a camera jointly monitor the crowd entering a stadium. The main contributions of this work are as follows:

1. **Perimeter defense problem with heterogeneous defenders:** We address a perimeter defense problem in a conical environment with two cooperative heterogeneous defenders, i.e., a vehicle and a turret, tasked to defend a perimeter. The vehicle has a finite capture radius and moves with unit speed, whereas the turret has a finite engage-

ment range and turns in the environment with a fixed angular rate. We do not impose any assumption on the arrival process of the intruders. More precisely, an arbitrary number of intruders can be released in the environment at arbitrary locations and time instances. Upon release, the intruders move with fixed speed $v$ towards the perimeter. Thus, the perimeter defense problem is characterized by six parameters: (i) angle $\theta$ of the conical environment, (ii) the speed $v$ of the intruders, (iii) the perimeter radius $\rho$, (iv) the engagement range of the turret $r_t$, (v) the angular rate of the turret $\omega$, and (vi) the capture radius of the vehicle $r_c$.

2. **Necessary condition:** We establish a necessary condition on the existence of any $c$-competitive algorithm for any finite $c$. This condition serves as a fundamental limit to this problem and identifies regimes for the six problem parameters in which this problem does not admit an effective online algorithm.

3. **Algorithm Design and Analysis:** We design and analyze four classes of cooperative algorithms with provably finite competitive ratios under specific parameter regimes. Specifically, the first two cooperative algorithms are provably 1-competitive, the third cooperative algorithm exhibits a finite competitive ratio which depends on the problem parameters and finally, the fourth algorithm is 1.5-competitive.

Additionally, through multiple parameter regime plots, we shed light into the relative comparison and the effectiveness of our algorithms. We also provide a brief discussion on the time complexity of our algorithms and how this work can be extended to other models of the vehicle.

This chapter is organized as follows. In Section 5.1, we formally define the competitive ratio and our problem. Section 5.2 establishes the necessary conditions, Section 5.3 presents the algorithms and their analysis. Section 5.4 provides several numerical insights and Section 5.5 discusses the time complexity of our algorithms and provides insights into how this work

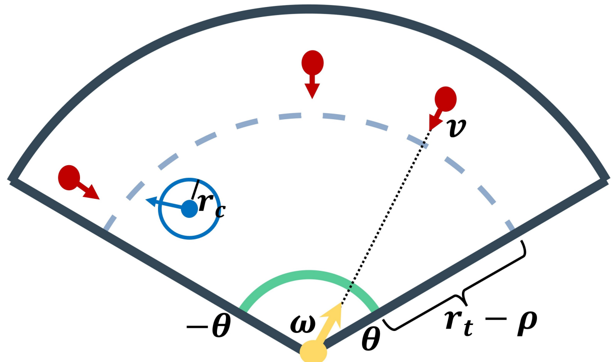



Figure 5.1 Problem Description. The vehicle is depicted by a blue dot and the blue circle around the vehicle depicts the capture circle. The direction of the vehicle is shown by the blue arrow. The yellow arrow depicts the turret (located at the origin of $\mathcal{E}(\theta)$) and the blue dashed curve denotes the engagement range of the turret. The green curve denotes the perimeter and the red dots denote the intruders. Note that the intruder that the turret is pointing to (black dashed line) is not captured unless it is within the engagement range of the turret (blue dashed curve).

extends to defenders with different motion models. Finally, Section 5.6 summarizes the chapter and outlines future directions.

## 5.1 Problem Formulation

Consider a planar conical environment (Figure 5.1) described by $\mathcal{E}(\theta) = \{(y, \alpha) : 0 < y \leq 1, -\theta \leq \alpha \leq \theta\}$, where $(y, \alpha)$ denotes a location in polar coordinates. The environment has two endpoints, $(1, \theta)$ and $(1, -\theta)$. The environment contains a concentric and coaxial region, $\mathcal{R}$, described by a set of points $(z, \alpha)$ in polar coordinates, where $0 < z \leq \rho$ and $\alpha \in [-\theta, \theta]$. Mathematically, $\mathcal{R}(\rho, \theta) = \{(z, \alpha) : 0 < z \leq \rho < 1, -\theta \leq \alpha \leq \theta\}$ for some $\rho \in (0, 1)$. Analogous to the environment, $\mathcal{R}$'s endpoints are $(\rho, \theta)$ and $(\rho, -\theta)$. Arbitrary numbers of intruders are released at the circumference of the environment, i.e., $y = 1$, at arbitrary time instants. Upon release, each intruder moves radially inward with a fixed speed $v > 0$[1] toward the perimeter $\partial\mathcal{R}(\theta) = \{(\rho, \alpha) : -\theta \leq \alpha \leq \theta\}$. Mathematically, if the $i$th intruder is released at time $t_i$, then its location is represented by a constant angle $\theta_i$ and its distance $z_i^t$ from the origin satisfying $z_i^t = 1 - v(t - t_i), \forall t \in [t_i, t_i + (1 - \rho)/v]$. Two defenders are employed to defend the perimeter, $\partial\mathcal{R}$ of the region $\mathcal{R}(\rho, \theta)$: a turret located

[1] As the speed of the intruders is normalized by the speed of the vehicle, we use the speed of intruders and speed ratio interchangeably.

at the origin of $\mathcal{E}(\theta)$ and a vehicle, both with simple motion dynamics. The vehicle has a finite capture radius, $r_c > 0$ and can either move with unit speed or remain stationary. The turret has a finite engagement range, $r_t$ such that $\rho \leq r_t \leq 1$, and can either turn clockwise or anti-clockwise with an angular speed of at most $\omega$ or remain stationary. We consider that the vehicle's capture radius is sufficiently small, in particular, $r_c < \min\{\rho, \rho\tan(\theta)\}$. Otherwise, this problem becomes trivial (refer to [10]).

Intruder $i$, located at $(z_i^t, \theta_i)$, is said to be captured at time instant $t$ if either one of the following holds:

- intruder $i$ is inside or on the capture circle of the mobile vehicle at time $t$, or
- intruder $i$ is at most $r_t$ distance away from the origin and $\gamma_t = \theta_i$ holds, where $\gamma_t$ denotes the heading of the turret at time instant $t$.

The intruder is said to be lost by the defenders if it reaches the perimeter without getting captured. The intruder is removed from $\mathcal{E}(\theta)$ if it is either captured or lost. We assume that the turret and the vehicle neutralize an intruder instantaneously, i.e., they do not require any additional service time. This implies that the defenders do not need to stop to complete the capture of an intruder. We further assume that the turret can start and stop firing instantaneously. This implies that the turret does not neutralize the vehicle in case the turret's heading angle is the same as the vehicle's angular coordinate at a particular time instant. It is worth highlighting that this work extends, with minor modifications, to when turret requires a finite amount of spool-up time. We refer to [7] where a turret with finite service time is considered.

A *problem instance* $\mathcal{P}$ is characterized by six parameters: (1) the speed of the intruders $v > 0$, (2) the perimeter radius $0 < \rho < 1$, (3) the angle $0 < \theta \leq \pi$ that defines the size of the environment as well as the perimeter, (4) the capture radius of the vehicle $0 < r_c < \min\{\rho, \rho\tan(\theta)\}$, (5) the angular speed of the turret $\omega > 0$, and (6) the range of the turret $\rho \leq r_t \leq 1$. We now formally define the objective of this work.

**Problem Statement:** *Design online deterministic cooperative algorithms with finite competitive ratios for the defenders and establish fundamental guarantees on the existence of online algorithms with finite competitive ratio.*

We start by determining a fundamental limit on the existence of $c$-competitive algorithms followed by designing online cooperative algorithms.

## 5.2 Fundamental Limits

We start by defining a *partition* of the environment. A partition of $\mathcal{E}(\theta)$ is a collection of $q \geq 1$ cones $\mathcal{W} = \{W_1, W_2, \ldots, W_q\}$ with disjoint interiors and whose union is $\mathcal{E}(\theta)$. Additionally, each cone is of unit radius having a finite positive angle and is concentric with the environment. We refer to a cone $W_m, 1 \leq i \leq q$ as the $m$th dominance region. Further, an endpoint of a dominance region is defined analogously as the endpoints of the environment. Given any set of initial locations of the defenders with distinct angular coordinates, the environment $\mathcal{E}(\theta)$ can be partitioned into two dominance regions such that each dominance region corresponds to a particular defender. We denote the portion of the perimeter contained in dominance region $m, 1 \leq m \leq 2$, as $\partial R_m$. Without loss of generality, we assume that $\partial \mathcal{R}_1$ (resp. $\partial \mathcal{R}_2$) corresponds to the leftmost (resp. rightmost) dominance region.

Let $f_1(\alpha) = 2\omega(\rho \sin(\alpha) - r_c) + \alpha - \theta$ and $f_2(\alpha) = 0.5\omega(\rho \sin(\alpha) - r_c) + \alpha - \theta$. Let $\bar{\alpha}_1$ denote the solution to the equation $f_1(\alpha) = 0$ and let $\bar{\alpha}_2$ be the solution to the equation $f_2(\alpha) = 0$. Then, following result establishes a necessary condition on the existence of $c$-competitive algorithms.

**Theorem 50** (Necessary Condition for finite $c_{\mathcal{A}}(\mathcal{P})$)**.** *Let*

$$\alpha_1^* = \begin{cases} \theta - 2\omega(\rho - r_c), & \textit{if } \theta \geq \frac{\pi}{2} + 2\omega(\rho - r_c), \\ \bar{\alpha}_1, & \textit{otherwise,} \end{cases}$$

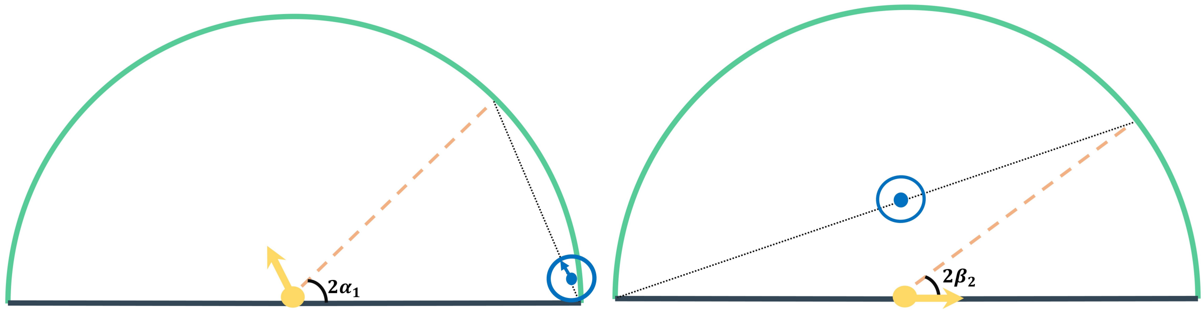
*and let*

$$\alpha_2^* = \begin{cases} \theta - 0.5\omega(\rho - r_c), & \textit{if } \theta \geq \frac{\pi}{2} + \omega(\rho - r_c), \\ \bar{\alpha}_2, & \textit{otherwise.} \end{cases}$$

*Then, for any problem instance $\mathcal{P}$ such that $\rho\sin(\theta) > r_c$ and $\frac{v}{\omega} > \frac{1-\rho}{\min\{\theta-\alpha_1^*, 2(\theta-\alpha_2^*)\}}$ holds, there does not exist a c-competitive algorithm for any finite c.*

*Proof.* Recall from Definition 1 that any online algorithm $\mathcal{A}$ is $c$-competitive if the condition $n_{\mathcal{O}}(\mathcal{I},\mathcal{P}) \leq cn_{\mathcal{A}}(\mathcal{I},\mathcal{P})$ holds for *every* input sequence $\mathcal{I}$. Thus, the aim is to construct an input sequence $\mathcal{I}$ such that the condition $n_{\mathcal{O}}(\mathcal{I},\mathcal{P}) \leq cn_{\mathcal{A}}(\mathcal{I},\mathcal{P})$ does not hold for any constant $c \geq 1$ regardless of which online algorithm is used. The proof is in three parts. First, we construct an input sequence $\mathcal{I}$. Then, we determine the best locations for the defenders against such an input sequence. Finally, we evaluate the performance of any online algorithm $\mathcal{A}$ on the input $\mathcal{I}$ as well as the performance of the optimal offline algorithm $\mathcal{O}$ on the same input sequence $\mathcal{I}$, to establish the result. Without loss of generality, we assume that both $\mathcal{A}$ and $\mathcal{O}$ have the vehicle at the origin at time instant 0 and the turret at angle $\gamma_0 = 0$.

Let $\mathbf{I} = \{\mathcal{I}_1, \mathcal{I}_2\}$ denote a set of two input sequences. Each input sequence $\mathcal{I}_l \in \mathbf{I}$, where $l \in \{1,2\}$, differs in the location of the arrival of intruders. Both input sequences $\mathcal{I}_1$ and $\mathcal{I}_2$ start at time instant $\max\{1, \frac{\theta}{\omega}\}$ and consists of a stream of intruders, i.e., a sequence of a single intruder arriving at location $(1,\theta)$ at every time instant $\max\{1, \frac{\theta}{\omega}\} + k\frac{1-\rho}{v}, k \in \mathbb{N} \cup \{0\}$ until time instant $t$. The time instant $t \geq 0$ corresponds to the time instant when either the vehicle or the turret, following any online algorithm $\mathcal{A}$, captures an intruder from the stream. A burst of $c+1$ intruders are then released at time instant $t$. The location where the burst of intruder arrives is different for each input sequence $\mathcal{I}_l, l \in \{1,2\}$. Given the location of the turret and the vehicle at time instant $t$, there can be at most two dominance regions of the environment and thus, at most two locations where the burst of intruders can arrive. These locations have the same angular coordinate as the endpoints of each $\partial R_m, \forall m \in \{1,2\}$ excluding $\theta$ and including $-\theta$. Without loss of generality, the burst of intruders are released at location $(1,-\theta)$ for $\mathcal{I}_1$. Further, if the heading angle of the turret is the same as the angular coordinate of the vehicle at time $t$, i.e., $\gamma_t = \theta$ and the vehicle's angular coordinate is $\theta$ at time instant $t$, then the burst intruders arrive at $(1,-\theta)$ for both $\mathcal{I}_1$ and $\mathcal{I}_2$ (In this case, $\mathcal{I}_1$ is same as $\mathcal{I}_2$).




(a) Vehicle captures the $i$th intruder (Case 1). (b) Turret captures the $i$th intruder (Case 2).

Figure 5.2 Depiction the perimeter depicting the two cases under which one of the defenders captures the $i$th intruder.

If neither the vehicle nor the turret captures the stream intruder, the stream never ends and the result follows as the optimal offline algorithm $\mathcal{O}$ can have its vehicle move, at time instant 0, to location $(\rho, \theta)$ and capture all stream intruders. Thus, in the remainder of the proof, we only consider online algorithms $\mathcal{A}$ for which either the vehicle or the turret captures at least one stream intruder at time instant $t$. Since the stream intruders arrive every $\frac{1-\rho}{v}$ time units apart and stops when an intruder from the stream is captured, it follows that no online algorithm can capture more than one intruder from the stream. Thus, we assume that the $i$th stream intruder was captured at time instant $t$, for some $i \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of positive integers.

We now determine the best locations, or equivalently the dominance regions of the environment, for the turret and the vehicle at time instant $t$. Note that the heading angle of the turret must not be equal to the angular coordinate of the vehicle at time instant $t$. This is because in such case, the burst arrives at $(1, -\theta)$ and thus, there always exist a location closer to angle $-\theta$ such that the vehicle or the turret can reach angular location $-\theta$ in less time. This implies that at time instant $t$, the environment consists of two dominance regions, each of which contains a defender. We denote the dominance region which contains the vehicle (resp. turret) as $W_{\text{Veh}}$ (resp. $W_{\text{Tur}}$) and determine them in the following two cases. These two cases arise based on whether the vehicle or the turret captures the $i$th intruder, each of which is considered below.

**Case 1 (Vehicle captures the $i$th intruder):** Let $2\alpha_1$ and $2\beta_1$ be the angles of $W_{\text{Veh}}$ and $W_{\text{Tur}}$, respectively. The best location for the vehicle and the turret in this case can be summarized as follows. The vehicle must be located on the line joining the two endpoints of the perimeter within its $W_{\text{Veh}}$ ($\partial\mathcal{R}_2$) only if $2\alpha_1 < \pi$. Otherwise, vehicle must be located on the line joining the origin to the location $(1, \theta)$. In both cases, it must be at a distance $r$ from location $(\rho, \theta)$. The angle of the turret must be equal to the angle bisector of $2\beta_1$ (Figure 5.2a). Finally, the time taken by the vehicle to reach the other endpoint of $\partial\mathcal{R}_2$ must be equal to the time taken by the turret to turn to the same angle corresponding to that location. This is denoted mathematically as

$$\frac{\theta - \alpha_1}{\omega} = \begin{cases} 2(\rho \sin(\alpha_1) - r_c) \text{ if } \alpha_1 < \frac{\pi}{2} \\ 2(\rho - r_c), \text{ otherwise} \end{cases} \tag{5.1}$$

where by definition $2\theta = 2\alpha_1 + 2\beta_1$ and the time taken by the vehicle to capture intruders at the other endpoint of the perimeter contained in its dominance region is $2(\rho\sin(\alpha_1) - r_c)$ (resp. $2(\rho - r_c)$) when $\alpha_1 < \frac{\pi}{2}$ (resp. $\alpha_1 \geq \frac{\pi}{2}$). As $\frac{\theta - \alpha_1}{\omega} = 2(\rho - r_c)$ only holds if $\alpha_1 \geq \frac{\pi}{2}$, it follows that $\alpha_1^* = \theta - 2\omega(\rho - r_c)$ if $\theta \geq \frac{\pi}{2} + 2\omega(\rho - r_c)$ holds. Otherwise, $\alpha_1^*$ is determined by solving the equation $f_1(\alpha) = 0$, where $f_1(\alpha) = 2\omega(\rho\sin(\alpha) - r_c) + \alpha - \theta$ where $\alpha_1 \in [0, \pi/2)$. We now show that the solution to $f_1(\alpha) = 0$ always exist if $r_c < \rho\sin(\theta)$.

Suppose that $\alpha_1 = \epsilon$, where $\epsilon > 0$ is a very small number. Then, $f(\epsilon) = 2\omega\rho\sin(\epsilon) + \epsilon - \theta - 2r_c\omega < 0$. Now consider that $\alpha_1 = \theta - \epsilon$ for the same $\epsilon$. Then, as $\rho\sin(\theta) > r_c$, it follows that $f_1(\theta - \epsilon) = 2\omega(\rho\sin(\theta - \epsilon) - r_c) - \epsilon > 0$, for a sufficiently small $\epsilon$. This means that for a sufficiently small $\epsilon > 0$, $f_1(\cdot)$ changes its sign in the interval $[\epsilon, \theta - \epsilon]$. Thus, from Intermediate Value Theorem and using the fact that $f_1(\alpha_1)$ is continuous function of $\alpha_1$, it follows that there must exist an $\alpha_1^*$ such that $f_1(\alpha_1^*) = 0$ if $r_c < \rho\sin(\theta)$. Further, since $f_1(\alpha)$ is a continuous function and its derivative is strictly increasing for $\alpha \in [0, \frac{\pi}{2})$ and hence, there exists a unique $\bar{\alpha}_1 \in [0, \frac{\pi}{2})$ which satisfies $f_1(\alpha)$.

**Case 2 (Turret captures the $i$th intruder):** Similar to Case 1, let $2\alpha_2$ and $2\beta_2$ be the angles of $W_{\text{Veh}}$ and $W_{\text{Tur}}$, respectively (Figure 5.2b). As the turret captures the stream intruder, it follows that $\gamma_t = \theta$. Further, the vehicle must be located at the midpoint of the line joining the two endpoints of the perimeter within its dominance region. Finally, the time taken by the vehicle to reach any endpoint of the perimeter must be equal to the time taken by the turret to turn to the same angle corresponding to that location. Mathematically, this yields

$$\frac{2(\theta - \alpha_2)}{\omega} = \begin{cases} \rho\sin(\alpha_2) - r_c, & \text{if } \alpha_2 < \frac{\pi}{2} \\ (\rho - r_c), & \text{otherwise} \end{cases} \tag{5.2}$$

where we used the fact that $2\theta = 2\alpha_2 + 2\beta_2$ and $\rho\sin(\alpha_2) - r_c$ (resp. $(\rho - r_c)$) denotes the time taken by the vehicle to capture intruders at the other endpoint of the perimeter contained in its dominance region when $\alpha_2 < \frac{\pi}{2}$ (resp. $\alpha_2 \geq \frac{\pi}{2}$). As $2\frac{\theta-\alpha_2}{\omega} = (\rho - r_c)$ only holds if $\alpha_2 \geq \frac{\pi}{2}$, it follows that $\alpha_2^* = \theta - 0.5\omega(\rho - r_c)$ if $\theta \geq \frac{\pi}{2} + 0.5\omega(\rho - r_c)$ holds. Otherwise, $\alpha_2^*$ is determined by solving the equation $f_2(\alpha) = 0$, where $f_2(\alpha) = 0.5\omega(\rho\sin(\alpha) - r_c) + \alpha - \theta$. By following similar steps as in Case 1, it can be shown that a unique solution to $f_2(\alpha) = 0$ always exists if $r_c < \rho\sin(\theta)$ and thus, has been omitted for brevity.

As $\frac{1-\rho}{v} < \min\{2(\rho\sin(\alpha_1^*) - r_c), \rho\sin(\alpha_2^*) - r_c\}$ or equivalently $v > \frac{1-\rho}{\min\{2(\rho\sin(\alpha_1^*) - r_c), \rho\sin(\alpha_2^*) - r_c\}}$ holds for $\alpha_1^* < \frac{\pi}{2}$, it follows that the vehicle cannot capture the burst intruders from $\mathcal{I}_2$ or $\mathcal{I}_1$. Further, as $2(\rho\sin(\alpha_1^*) - r_c) = \frac{\theta - \alpha_1^*}{\omega}$ or $\rho\sin(\alpha_2^*) - r_c = \frac{2(\theta - \alpha_2^*)}{\omega}$ holds at time instant $t$, it follows that the turret can also not capture the burst intruders from both $\mathcal{I}_1$ and $\mathcal{I}_2$. Therefore, the turret and the vehicle jointly captures at most one intruder from input instance $\mathcal{I}_1$ as well as $\mathcal{I}_2$. A similar conclusion holds when $\alpha_1^* \geq \frac{\pi}{2}$ or when $\alpha_2^* \geq \frac{\pi}{2}$.

Thus, we have established that for any online algorithm $\mathcal{A}$, the vehicle and the turret can jointly capture at most a single intruder from input instance $\mathcal{I}_l, 1 \leq l \leq 2$. We now show that the optimal offline algorithm $\mathcal{O}$ captures all of the intruders on the input sequence $\mathcal{I}_l, 1 \leq l \leq 2$.

Recall that $\mathcal{O}$ has complete information at time 0, of when, where, and how many intruders will arrive. Thus, at time 0, $\mathcal{O}$ moves its vehicle to location $(\rho, \theta)$ and the turret to angle $-\theta$. The defenders of $\mathcal{O}$ have sufficient time to reach these locations as the first intruder arrives at time instant $\max\{1, \frac{\theta}{\omega}\}$ and thus, the capture of all $i$ stream intruders as well as the burst intruders is guaranteed. Thus, $n_{\mathcal{O}}(\mathcal{I}, \mathcal{P}) = i + c + 1$ and $n_{\mathcal{A}}(\mathcal{I}, \mathcal{P}) = 1$ which yields that $\frac{n_{\mathcal{O}}(\mathcal{I},\mathcal{P})}{n_{\mathcal{A}}(\mathcal{I},\mathcal{P})} = i + c + 1$. As $i + c + 1 > c$ for any constant $c$, it follows that $n_{\mathcal{O}}(\mathcal{I}) \leq c n_{\mathcal{A}}(\mathcal{I})$ does not hold for any $c$. This concludes the proof. □ □

**Remark 9.** *Since we do not impose any restriction on the number of intruders that can arrive in the environment, an adversary can repeat the input sequence designed in the proof of Theorem 50 any number of times. Thus, the lower bound derived in Theorem 50 holds asymptotically for when $\bar{c} > 0$ in the definition of c-competitive algorithms as well.*

We now turn our attention to designing online algorithms and deriving upper bounds on their competitive ratios. In the next section, we design and analyze four online algorithms, each with a provably finite competitive ratio in a specified parameter regime.

### 5.3 Online Algorithms

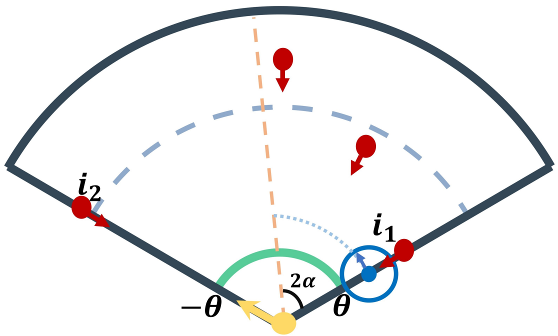
In this section, we design and analyze online algorithms and characterize the parameter space in which these algorithms have finite competitive ratios. The parameter space of all of our algorithms is characterized by two main quantities:

- The time taken by the intruders to reach the perimeter in the worst-case and
- the time taken by the defenders to complete their respective motions.

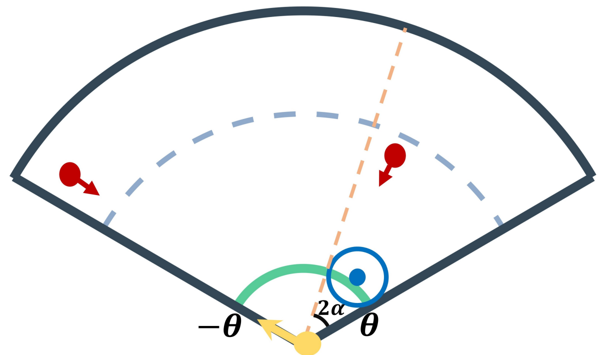
Intuitively, the parameter space is obtained by comparing the aforementioned quantities[2]. Since, the time taken by the intruders is inversely proportional to $v$, the $(v, \rho)$ parameter space of our algorithms can be increased by designing the algorithms such that the time taken by the defenders to complete their motions is the least. We characterize the time taken by the defenders to complete their respective motions as *epochs*, which is formally defined as follows.

An epoch $k$ for an algorithm is defined as the time interval which begins when at least one of the two defenders moves from its initial position and ends when both defenders return

[2]the time taken by the defenders must be at most the time taken by the intruders to reach the perimeter




(a) Configuration of the defenders for Algorithm SiR for Case 1 ($\alpha = \frac{\theta}{1+\omega z_v}$). The blue dot curve depicts the path taken by the vehicle for angular motion.




(b) Configuration of the defenders for Algorithm SiR for Case 2 ($\alpha = \arctan(r_c/\rho)$).

Figure 5.3 Partition (shown by the orange line) of the environment and the configuration of the defenders for the two cases of Algorithm SiR.

back to their respective initial positions. We denote the time instant when an epoch $k$ begins as $k_s$.

We now describe our first algorithm that has the best possible competitive ratio.

### 5.3.1 Sweep within Dominance Region (SiR)

Algorithm SiR is an open-loop and memoryless algorithm in which we constrain the vehicle to move in an angular motion, i.e., either clockwise or anti-clockwise. This can be achieved by moving the vehicle with unit speed in the direction perpendicular to its position vector (see Figure 5.3a). By doing so, we aim to understand the worst-case scenarios for heterogeneous defenders and gain insights into the effect of the heterogeneity that arises due to the capture range of the defenders, i.e., the capture circle and the engagement range of the turret. We say that a defender *sweeps* in its dominance region if it turns, from its starting location, either clockwise or anti-clockwise to a specified location and then turns back to its initial location. Algorithm SiR is formally defined in Algorithm 9 and is summarized as follows.

Algorithm SiR first partitions the environment $\mathcal{E}(\theta)$ into two dominance regions and assigns a single defender to a dominance region. Let $2\alpha$ denote the angle of $W_{\text{Veh}}$. Then, the vehicle takes exactly $4\alpha z_v$ to sweep within its dominance region, where $z_v$ denotes the radial

**Algorithm 9:** Sweep within Dominance Region Algorithm

Turret is at angle $-\theta$<br>
**if** $\alpha = \arctan(\frac{r_c}{\rho})$ **then**<br>
    Vehicle is located at $(\frac{\rho}{cos(\alpha)}, \theta - \alpha)$<br>
    **for** *each epoch* $k$ **do**<br>
        Turn the turret clockwise to angle $\theta - 2\alpha$<br>
        Turn the turret anti-clockwise to angle $-\theta$<br>
    **end**<br>
**else**<br>
    Vehicle is located at $(z_v, \theta)$<br>
    **for** *each epoch* $k$ **do**<br>
        Turn vehicle (resp. turret) anti-clockwise (resp. clockwise) until it reaches location (resp. angle) $(z_v, \theta - 2\alpha)$ (resp. $\theta - 2\alpha$)<br>
        Turn vehicle (resp. turret) clockwise (resp. anti-clockwise) until it reaches location (resp. angle) $(z_v, \theta)$ (resp. $-\theta$)<br>
    **end**<br>
**end**

location of the vehicle and will be determined shortly (see Case 1 below). Similarly, as the turret can only turn either clockwise or anti-clockwise with at most angular speed $\omega$, the turret takes exactly $\frac{4(\theta-\alpha)}{\omega}$ to sweep in its respective dominance region $W_{\text{Tur}}$. Observe that the environment must be partitioned such that the time taken by the defenders to complete their motion in their respective dominance region is equal. Otherwise, in the worst-case, all of the intruders will be concentrated in the dominance region of that defender that takes more time to sweep its dominance region. Mathematically, this means $4\alpha z_v = \frac{4(\theta-\alpha)}{\omega}$ must hold which yields that $\alpha = \frac{\theta}{1+\omega z_v}$. Observe that as $\omega \to \infty$, $\alpha \to 0$. This means that the turret sweeps the entire environment, in time $\frac{4\theta}{\omega}$, if $\omega$ is sufficiently high. Recall that the $(v, \rho)$ parameter space is characterized by the time taken by the defenders to complete their motion and can be improved by reducing the time taken by the turret, for high values of $\omega$. In case of very high $\omega$, this can be achieved by having the vehicle remain static at a specific location while the turret sweeps the remaining environment as opposed to the entire environment. This means that although angle $\alpha = \frac{\theta}{1+\omega z_v}$ characterizes the vehicle's dominance region, there exists an angle $\hat{\alpha} \geq \alpha$ for some values of problem parameters for which we can obtain an improved parameter regime by assigning a dominance region of angle $2\hat{\alpha}$ to the vehicle. Thus, in Algorithm SiR, there are two cases based on the values of the problem parameters. First, as described above, the defenders sweep the environment in their respective dominance regions and second, the vehicle remains static at a specific location while the turret sweeps its dominance region. In what follows, we determine the location at which the vehicle must remain static for the second case, followed by formally describing the two cases.

The vehicle's location must be such that its capture circle covers the perimeter contained in its dominance region entirely, ensuring that any intruder that arrives in that dominance region is guaranteed to be captured. To achieve this, the boundary of the dominance region assigned to the vehicle must be tangent to its capture circle (see Figure 5.3b) which, through geometry, yields that $\hat{\alpha} = \arctan \frac{r_c}{\rho}$ and the location for the vehicle as $(\frac{\rho}{\cos(\hat{\alpha})}, \theta - \hat{\alpha})$. Therefore, the angle of the vehicle's dominance region is defined as $2\alpha = 2\max\{\frac{\theta}{1+\omega z_v}, \hat{\alpha}\}$, where $\hat{\alpha} = \arctan \frac{r_c}{\rho}$, and angle $\alpha$ determines if the vehicle sweeps in its dominance region or remains stationary. We first describe the motion of the turret followed by formally describing the motion of the vehicle in the two cases.

At time instant 0, the turret is at an angle $-\theta$. The turret turns clockwise, with angular speed $\omega$ towards angle $\theta - 2\alpha$. Upon reaching angle $\theta - 2\alpha$, the turret turns anti-clockwise towards angle $-\theta$. Note that the turret takes exactly $\frac{4(\theta-\alpha)}{\omega}$ time to complete its motion in a particular epoch. We now describe the motion of the vehicle which can be summarized in two cases described as follows:

**Case 1 ($\alpha = \frac{\theta}{1+\omega z_v}$):** At time instant 0, the vehicle is located at $(z_v, \theta)$, where $z_v = \min\{\rho + r_c, 1 - r_c\}$ and was determined in [10] and was proved to be optimal in Chapter 4. The vehicle then moves anti-clockwise with unit speed in the direction perpendicular to its position vector until it reaches the location $(z_v, \theta - 2\alpha)$. Then, the vehicle moves clockwise, with direction perpendicular to its position vector, until it reaches location $(z_v, \theta)$. Note that the vehicle takes exactly $4\alpha z_v$ time to complete its motion in a particular epoch. Since $\alpha$

is chosen so that $4\alpha z_v = \frac{4(\theta-\alpha)}{\omega}$, the vehicle and the turret return to their respective initial locations at the same time instant, at which the next epoch begins.

**Case 2 ($\alpha = \arctan(\frac{r_c}{\rho})$):** At time instant 0, the vehicle is located at $(z_v = \frac{\rho}{\cos(\alpha)}, \theta - \alpha)$ and remains stationary at this location for the entire duration. In this case, the next epoch begins once the turret turns back to angle $-\theta$.

The following result characterizes the parameter regime in which Algorithm SiR is 1-competitive.

**Theorem 51.** *Algorithm SiR is* 1*-competitive for a set of problem parameters that satisfy*

$$v \leq \begin{cases} \min\left\{\frac{(r_t-\rho)(1+\omega z_v)}{4\theta z_v}, \frac{(z_v+r_c-\rho)(1+\omega z_v)}{4\theta z_v}\right\} & \text{if } \frac{\theta}{1+\omega z_v} > \arctan(r_c/\rho) \\ \frac{(r_t-\rho)\omega}{4(\theta-\arctan(r_c/\rho))}, & \text{otherwise.} \end{cases}$$

*Otherwise, Algorithm SiR is not $c$-competitive for any constant $c$.*

*Proof.* Suppose that $\alpha = \frac{\theta}{1+\omega z_v}$. At the start of any epoch $k$, i.e., at time instant $k_s$, we assume that, in the worst-case, intruders $i_1$ and $i_2$ are located at $(z_v + r_c + \epsilon_1, \theta)$ and $(r_t + \epsilon_2, -\theta)$, respectively, where $\epsilon_1$ and $\epsilon_2$ are arbitrary small positive numbers (see Figure 5.3a). To ensure that the vehicle (resp. turret) does not lose any intruder, we require that the time taken by the vehicle (resp. turret) to return to location (resp. angle) $(z_v, \theta)$ (resp. $-\theta$) must be less than the time taken by intruder $i_1$ (resp. $i_2$) to reach the perimeter. Formally, $\frac{r_t+\epsilon_2-\rho}{v} \geq \frac{4(\theta-\alpha)}{\omega}$ and $\frac{z_v+r_c+\epsilon_1-\rho}{v} \geq 4\alpha z_v$ must hold. Given the first condition on $v$, these two conditions always hold, so any intruder that arrives in the environment is guaranteed to be captured.

If $v > \min\{\frac{(r_t-\rho)(1+\omega z_v)}{4\theta z_v}, \frac{(z_v+r_c-\rho)(1+\omega z_v)}{4\theta z_v}\}$, then there exists an input instance with intruders arriving only at $(1, -\theta)$ such that these intruders are located at $(r_t + \epsilon, -\theta)$ at the time instant the turret turns from angle $-\theta$. As $v > \min\{\frac{(r_t-\rho)(1+\omega z_v)}{4\theta z_v}, \frac{(z_v+r_c-\rho)(1+\omega z_v)}{4\theta z_v}\}$ all of these intruders will be lost and thus, from Definition 1, Algorithm SiR will not be $c$-competitive.

Now consider that $\alpha = \arctan(\frac{r_c}{\rho})$. As the vehicle remains stationary in its dominance region and the location of the vehicle is such that no intruder that is released in that dominance region can reach the perimeter, we only focus on the turret. Assume that, in the worst-case, intruder $i_1$ is located at $(r_t+\epsilon, -\theta)$ where $\epsilon$ is an arbitrary small positive numbers. To ensure that the turret does not lose any intruder, we require that the time taken by the turret to return to angle $-\theta$ must be less than the time taken by intruder $i_1$ perimeter. Given the second condition on $v$, i.e., $v \leq \frac{(r_t-\rho)\omega}{4(\theta-\arctan(r_c/\rho))}$ holds, it is ensured that intruder $i_1$ will be captured. The proof for Algorithm SiR not being $c$-competitive when $v > \frac{(r_t-\rho)\omega}{4(\theta-\arctan(r_c/\rho))}$ is analogous to the previous case and has been omitted for brevity. This concludes the proof. □

Although Algorithm SiR is 1-competitive, note that for $r_t = \rho$, the algorithm is not effective as Theorem 51 yields $v \leq 0$. However, by allowing the vehicle to sweep the entire environment, it is still possible to capture all intruders for some small $v > 0$. This is addressed in a similar algorithm below.

### 5.3.2 Sweep in Conjunction (SiCon)

At time instant 0, the turret is at angle $\theta$ and the vehicle is located at location $(\min\{r_t + r_c, 1\}, \theta)$. The idea is to move the two defenders together in angular motion. Thus, the vehicle moves anti-clockwise with unit speed in the direction perpendicular to its position vector until it reaches the location $(\min\{r_t + r_c, 1\}, -\theta)$. Similarly, the turret turns anti-clockwise, in conjunction with the vehicle, to angle $-\theta$. Upon reaching $-\theta$, the vehicle and the turret move clockwise until they reach angle $\theta$. The defenders then begin the next epoch. As the two defenders move in conjunction, $\frac{2\theta}{\omega} = 2\theta \min\{r_t + r_c, 1\} \Rightarrow \omega = \frac{1}{\min\{r_t+r_c,1\}}$ must hold. Thus, this algorithm is effective for $\omega \geq \frac{1}{\min\{r_t+r_c,1\}}$ by turning the turret exactly with angular speed $\frac{1}{\min\{r_t+r_c,1\}}$.

The following result establishes that Algorithm SiCon is 1-competitive for specific parameter regimes.

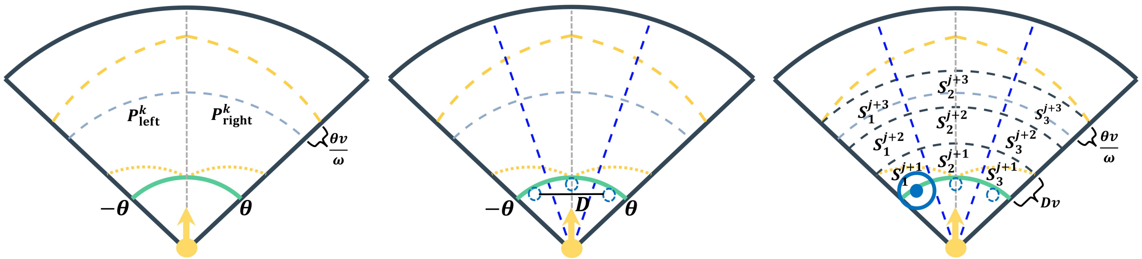
**Theorem 52.** *Algorithm SiCon is* 1*-competitive for a set of problem parameters which satisfy* $\omega \geq \frac{1}{\min\{r_t+r_c,1\}}$ *and* $v \leq \frac{\min\{r_t+2r_c,1\}-\rho}{4\theta\min\{r_t+r_c,1\}}$*. Otherwise, it is not c-competitive for any constant*

$c$.

*Proof.* As the proof is analogous to the proof of Theorem 51, we only provide an outline of this proof for brevity. In the worst-case, an intruder requires exactly $\frac{\min\{r_t+2r_c,1\}-\rho}{v}$ time to reach the perimeter whereas, the defenders synchronously require $4\theta\min\{r_t+r_c,1\}$ time to complete their motion in any epoch. Thus, as the time taken by the defenders must be at most the time taken by the intruders we obtain the competitive ratio. If the condition on $v$ does not hold, then by constructing an input analogous to the input in the proof of Theorem 51, it can be shown that Algorithm SiCon is not $c$-competitive. □ □

**Remark 10** (Maneuvering Intruders)**.** *As the analysis of the fundamental limit (Theorem 50), Algorithm SiR, and Algorithm SiCon are independent of the nature of motion of the intruders, the results of Theorem 50, Algorithm SiR, and Algorithm SiCon apply directly to the case of maneuvering or evading intruders.*

Recall that in Algorithm SiR, the idea was to partition the environment and assign a single defender in each dominance region. By doing so, we obtain valuable insight into the parameter regime wherein we are guaranteed to capture all intruders. However, we refrain from designing such algorithms in this work due to the following two reasons. First, such an algorithm requires that ratio of intruders captured by a defender to the total number of intruders that arrived in that corresponding defender's dominance region is equal for both defenders. Otherwise, since the adversary has the information of the entire algorithm, it will release more intruders in the dominance region of the defender that has the lower ratio, which determines the competitive ratio of such an algorithm. Second, such algorithms are not cooperative and thus fall out of the scope of this chapter. The objective of this work is to study how heterogeneous defenders can be used to improve the competitive ratio of a single defender. Thus, in the next algorithm, although we partition the environment, the defenders are not restricted to remain within their own dominance region.




(a) Description of sets $P^k_{\text{left}}$ and $P^k_{\text{right}}$. The grey dashed line denotes the splitting of the environment into two halves. The turret's heading angle is 0 at time instant $k_s$.

(b) Splitting the environment into $N = 3$ sectors. The blue dashed circles denote the resting points for the vehicle.

(c) Final representation of the environment for Algorithm Split. The black dashed curves denotes splitting of the environment radially corresponding to the three time intervals, each of length $D$. The vehicle is located at $(x_1, \phi_1)$.

Figure 5.4 Description of Algorithm Split. The light grey dashed line denotes the splitting of the environment. The region between the yellow dashed curve and the yellow dot curve on the left (resp. right) side of the grey line denotes $P^k_{\text{left}}$ (resp. $P^k_{\text{right}}$). The blue dashed straight lines denote the sectors ($N = 3$) of the environment and the black dashed curves denotes the three radial intervals of length $Dv$ each. The vehicle is located at $(x_1, \phi_1)$.

### 5.3.3 Split and Capture (Split)

The motivation for this algorithm is to utilize the vehicle's ability to move in any direction while the turret rotates either clockwise or anti-clockwise. Since the turret can only turn either clockwise or anti-clockwise, the idea is to first partition the environment into two halves and turn the turret towards the side which has higher number of intruders. By doing so, we hope to capture at least half of the intruders by the turret, assuming they are sufficiently slow, that arrive in every epoch. Further, while the turret moves to capture intruders on one side, the vehicle moves to the other side to capture intruders, ensuring that the defenders jointly capture more than half of the intruders that arrive in the environment in every epoch. Algorithm Split is formally defined in Algorithm 10 and is summarized as follows where we first describe the motion of the turret in every epoch followed by that of the vehicle.

The heading angle of the turret is always $\gamma_{k_s} = 0$ at the start of every epoch $k$. To determine whether the turret turns clockwise or anti-clockwise at time instant $k_s$, we first

**Algorithm 10:** Split and Capture Algorithm

Turret is at angle 0
Vehicle is located at $(x_1, \phi_1)$
**for** *each epoch k* **do**
  Compute $P^k_{\text{right}}$, $P^k_{\text{left}}$, $N_{p^*}$
  **if** $|P^k_{\text{right}}| \geq |P^k_{\text{left}}|$ **then**
    Turn the turret clockwise to angle $\theta$
    Turn the turret back to angle 0
  **else**
    Turn the turret anti-clockwise to angle $-\theta$
    Turn the turret back to angle 0
  **end**
  **if** $N_{p^*} \neq N_i$ **then**
    **if** $|S^{j+2}_{p^*}| \geq |S^{j+1}_i|$ **then**
      Move vehicle to $(x_{p^*}, \phi_{p^*})$ and then capture intruders in $S^{j+2}_{p^*}$
    **else**
      Keep vehicle at $(x_i, \phi_i)$ and capture intruders in $S^{j+1}_i$
      Move vehicle to $(x_{p^*}, \phi_{p^*})$
    **end**
  **else**
    Keep vehicle at $(x_i, \phi_i)$ and capture intruders in $S^{j+1}_i$ and $S^{j+2}_i$
  **end**
**end**

describe two sets $P_{\text{left}}$ and $P_{\text{right}}$. These sets characterize a region on the left and right side of the $y$-axis, respectively, and are determined once at time instant 0.

$$P_{\text{right}}(\rho, v) := \{(y, \beta) : \rho + \frac{\beta v}{\omega} < y \leq \min\{1, r_t + \frac{(2\theta - \beta)v}{\omega}\}, \quad \forall \beta \in [0, \theta]\},$$
$$P_{\text{left}}(\rho, v) := \{(y, \beta) : \rho - \frac{\beta v}{\omega} < y \leq \min\{1, r_t + \frac{(2\theta + \beta)v}{\omega}\}, \quad \forall \beta \in (0, -\theta]\}.$$

Let $P^k_{\text{right}}$ and $P^k_{\text{left}}$ denote the set of intruders contained in $P_{\text{right}}$ and $P_{\text{left}}$ (see Figure 5.4a), respectively, at the start of an epoch $k$ and let $|S|$ denote the cardinality of a set $S$ of intruders. Then, at time instant $k_s$, the turret compares the total number of intruders in $P^k_{\text{left}}$ to the total number of intruders in $P^k_{\text{right}}$ and turns in the direction of the set which has higher number of intruders. More formally, if $|P^k_{\text{right}}| < |P^k_{\text{left}}|$ holds at time instant $k_s$, then the turret turns anti-clockwise towards angle $-\theta$. Upon turning to angle $-\theta$, the turret turns to angle 0. Otherwise, i.e., if $|P^k_{\text{right}}| \geq |P^k_{\text{left}}|$ holds at time instant $k_s$, then the turret turns clockwise towards angle $\theta$. Upon turning to angle $\theta$, the turret turns to angle 0. As the turret's dominance region is determined at the start of every epoch $k$, we denote the turret's dominance region as $W^k_{\text{Tur}}$, and consequently, the other dominance region as the *vehicle's dominance region* denoted as $W^k_{\text{Veh}}$. We now characterize the motion for the vehicle which builds upon the SNP algorithm designed in [10].

Algorithm Split further divides the environment $\mathcal{E}(\theta)$ into $N = \lceil \frac{\theta}{\theta_s} \rceil$ sectors, where $2\theta_s = 2\arctan(r_c/\rho)$ denotes the angle of each sector (see Figure 5.4b). The value $2\arctan(r_c/\rho)$ of the angle of each of the sectors is to ensure that the portion of the perimeter in a sector can be completely contained in the capture circle of the vehicle. This can be achieved by positioning the vehicle at *resting points* (see Figure 5.4b), which is a specific location in every sector and is formally defined as follows.

**Definition 53** (Resting points). *Let $N_l$ denote the $l$th sector, for every $l \in \{1, \ldots, N\}$ where $N_1$ corresponds to the leftmost sector. Then, a resting point $(x_l, \phi_l) \in \mathcal{E}(\theta)$ for sector $N_l$, is the location for the vehicle such that when positioned at $(x_l, \phi_l)$, the portion of the perimeter inside sector $N_l$ is contained completely within the capture circle of the vehicle. Mathematically, this is equivalent to*

$$(x_l, \phi_l) = \Big(\tfrac{\rho}{\cos(\theta_s)}, (l - \tfrac{N+1}{2})2\theta_s\Big).$$

Now, let $D$ denote the distance between the two resting points that are furthest apart in the environment (see Figure 5.4b). Formally,

$$D = \begin{cases} 2\frac{\rho}{\cos(\theta_s)}\sin((N-1)\theta_s), \text{ if } (N-1)\theta_s < \frac{\pi}{2}, \\ 2\frac{\rho}{\cos(\theta_s)}, \text{ otherwise.} \end{cases} \tag{5.3}$$

Observe that when $N = 1$, $D = 0$. This means that the vehicle captures all intruders that arrive in the environment by positioning itself to the unique resting point of the single sector.

Next, Algorithm Split radially divides the environment $\mathcal{E}(\theta)$ into three intervals of length $Dv$, corresponding to time intervals of time length $D$ each (see Figure 5.4c). Specifically, the

$j$th time interval for any $j > 0$ is defined as the time interval $[(j-1)D, jD]$. Note that this time interval is different than the epoch of the algorithm. Let $S_l^j$ denote the set of intruders that are contained in the $l$th, $l \in \{1, \ldots, N\}$, sector and were released in the $j$th interval. Then, at the start of each epoch, the motion of the vehicle is based on the following two steps: First, select a sector with the maximum number of intruders. Second, determine if it is beneficial to switch over to that sector. These two steps are achieved by two simple comparisons; **C1** and **C2** detailed below.

Suppose that the vehicle is located at the resting point of sector $N_i$ at the start of the $j$th epoch and let $\tilde{\mathbf{N}}$ denote the set of sectors in the vehicle's dominance region. Then, to specify the first comparison **C1**, we associate each sector $N_l \in \tilde{\mathbf{N}}$ with the quantity

$$\eta_i^l \triangleq \begin{cases} |S_l^{j+2}| + |S_l^{j+3}|, & \text{if } l \neq i, \\ |S_i^{j+1}| + |S_i^{j+2}| + |S_i^{j+3}|, & \text{if } l = i. \end{cases}$$

Note that $\eta_i^l$ is not defined for every sector in the environment. Instead, it is only defined for the sectors in $W_{\text{Veh}}^j$, which may contain $N_i$. Then, as the outcome of **C1**, Algorithm Split selects the sector $N_{p^*}$, where $p^* = \arg\max_{p \in \tilde{\mathbf{N}}} \eta_i^p$. In case there are multiple sectors with same number of intruders, then Algorithm Split breaks the tie as follows. If the tie includes the current sector $N_i$ (which is only possible if $N_i \in \tilde{\mathbf{N}}$ holds[3]), then Algorithm Split selects $N_i$. Otherwise, Algorithm Split selects the sector contained in $\tilde{\mathbf{N}}$ with the maximum number of intruders in the interval $j+2$, i.e., $p^* = \arg\max_{p \in \hat{\mathbf{N}}} |S_{\text{p}}^{j+2}|$, where $\hat{\mathbf{N}}$ denotes the set of sectors that have the same number of intruders. If this results in another tie, then this second tie is resolved by selecting the sector with the least index. Let the sector chosen as the outcome of **C1** be $N_o$.

We now describe comparison **C2** jointly with the motion of the vehicle in the following two points:

[3]This case arises when the Algorithm Split moves the turret in the same direction for at least two consecutive epochs.

- If the sector chosen as the outcome of **C1** is $N_o, o \neq i$, and the total number of intruders in the set $S_o^{j+2}$ is no less than the total number of intruders in $S_i^{j+1}$, then Algorithm Split moves the vehicle to $(x_o, \phi_o)$ arriving in at most $D$ time units. Then the vehicle waits at that location to capture all intruders in $S_o^{j+2}$. Otherwise, i.e., the total number of intruders in $S_o^{j+2}$ is less than $S_i^{j+1}$, then the vehicle stays at $(x_i, \phi_i)$ until it captures all of the intruders in $S_i^{j+1}$ and then moves to $(x_o, \phi_o)$ arriving in at most $D$ time units.

- If the sector chosen is $N_i$, then the vehicle stays at its current location $(x_i, \phi_i)$ for $2D$ time units capturing intruders in $S_i^{j+1}$ and $S_i^{j+2}$.

Note that the vehicle takes at most $2D$ time units in every case above. The vehicle then re-evaluates after $2D$ time. At time instant 0, the turret's heading angle is 0 and the vehicle is located at $(x_1, \phi_1)$. The first epoch begins when the first intruder arrives in the environment.

We now describe two key requirements for the algorithm. The first requirement is to ensure that the defenders start their individual motion in an epoch at the same time instant. Recall that the turret requires exactly $\frac{2\theta}{\omega}$ time to turn from its initial heading angle to either $\theta$ or $-\theta$, at time instant $k_s$, and turn back to its initial heading angle. On the other hand, the vehicle requires $2D$ time units to capture intruders in at least one interval. Thus, to ensure that the defenders begin their motion at the same time instant, $\frac{2\theta}{\omega} = 2D$ must hold, i.e., the speed of the turret must be at least $\frac{\theta}{D}$. The second requirement is to ensure that the algorithm has a finite competitive ratio. This is achieved by ensuring that any intruder that was not accounted for comparison by the defenders ( for instance intruders that are not in $P^k_{\text{left}}$ or $P^k_{\text{right}}$) in an epoch $k$, are accounted in epoch $k+1$. Our next result formally characterizes this requirement for the turret.

**Lemma 54.** *Any intruder with radial coordinate greater than* $\min\{1, r_t + \frac{(2\theta-\beta)v}{\omega}\},\ \forall \beta \in [0, \theta]$ *(resp.* $\min\{1, r_t + \frac{(2\theta+\beta)v}{\omega}\},\ \forall \beta \in (0, -\theta]$*) at time instant $k_s$ will be contained in the set $P^{k+1}_{\text{right}}$ (resp. $P^{k+1}_{\text{left}}$) at time instant $(k+1)_s$ if $v \leq \frac{\omega(r_t-\rho)}{2\theta}$ holds.*

*Proof.* Without loss of generality, suppose that $|P^k_{\text{left}}| \leq |P^k_{\text{right}}|$ holds at time instant $k_s$. Then, the total time taken by the turret to move towards $\theta$ and turn back to angle 0 is $\frac{2\theta}{\omega}$. In order for any intruder $i$ to not be captured in epoch $k$, in the worst-case, the intruder $i$ must be located at $(\min\{1, r_t\} + \epsilon, \theta)$, where $\epsilon$ is a very small positive number, by the time the turret reaches angle $\theta$. Note that $1 + \epsilon$ here means that the intruder is released after $\epsilon$ time at location $(1, \theta)$ after the turret's heading angle is $\theta$. Thus, in order to ensure that $i$ can be captured in epoch $k + 1$, the condition $\frac{2\theta}{\omega} \leq \frac{r_t - \rho}{v}$ must hold, where we used the fact that $r_t \leq 1$ and $\epsilon$ is a very small positive number. From the definition of $P^k_{\text{right}}$, if the intruder $i$ can be captured in epoch $k + 1$, then it follows that the intruder $i$ was contained in the set $P^{k+1}_{\text{right}}$ at the start of epoch $(k + 1)$. This concludes the proof. □ □

The proof of Lemma 54 is established in the worst-case scenario which is that an intruder, with angular coordinate $\theta$ (resp. $-\theta$), is located just above the range of the turret at the time instant when the turret's heading angle is $\theta$ (resp. $-\theta$) in an epoch $k$. This is because in an epoch $k$, the angle $\theta$ is the only heading angle that is visited by the turret once as the turret turns to all angles $\beta$ $\forall \beta \in [0, \theta)$ (resp. $[0, -\theta)$) twice; once when the turret turns to angle $\theta$ or $-\theta$ and second, when turret turns back to angle 0. This results in the following corollary, the proof of which is analogous to the proof of Lemma 54.

**Corollary 9.** *Suppose that the turret moves to capture intruders in $P^k_{\text{left}}$ (resp. $P^k_{\text{right}}$) in epoch $k$. Then, the intruders contained in $P^k_{\text{right}}$ (resp. $P^k_{\text{left}}$) with radial coordinate strictly greater than $r_t + \frac{v\theta}{\omega}$ at time instant $k_s$ will be considered for comparison at the start of epoch $(k + 1)$ if $v \leq \frac{\omega(r_t - \rho)}{2\theta}$.*

Recall that the adversary selects the release times and the locations of the intruders in our setup. Thus, with the information of the online algorithm, the adversary can release intruders such that all the intruders have their radial coordinates at most $r_t + \frac{\theta v}{\omega}$ and at angular location $\theta$ of $-\theta$ at the start of every epoch $k$, which is considered to be the worst-case scenario. This ensures that if the turret selects to turn towards $-\theta$ (resp. $\theta$) at time

$k_s$, then the turret cannot capture any intruder that was contained in $P^k_{\text{right}}$ (resp. $P^k_{\text{left}}$) in epoch $k+1$. As the idea is to have the vehicle capture these intruders, we require that the intruders must be sufficiently slow. This is explained in greater detail as follows.

For the vehicle, the requirement is that the intruders take at least $3D$ time units to reach the perimeter. This is to ensure that the vehicle can account for intruders that are very close to the perimeter at the start of an epoch. From Corollary 9, as the intruders with radial location greater than $r_t + \frac{\theta v}{\omega}$ are counted for comparison in next epoch by the turret, we require that these intruders must also be counted by the vehicle in the next epoch. This yields that $3D \leq \frac{\min\{1, r_t + \frac{\theta v}{\omega}\} - \rho}{v}$ which implies that either $v \leq \frac{1-\rho}{3D}$ or $v \leq \frac{r_t - \rho}{2D}$ must hold, where we used the fact that $\frac{2\theta}{\omega} = 2D$. Finally, as Lemma 54 requires that $v \leq \frac{r_t-\rho}{2D}$ must hold, the second requirement for Algorithm Split is that $v \leq \min\{\frac{1-\rho}{3D}, \frac{r_t-\rho}{2D}\}$. We now establish the competitive ratio of Algorithm Split.

**Theorem 55.** *Let* $\theta_s = \arctan(r_c/\rho)$ *and* $N = \lceil \frac{\theta}{\theta_s} \rceil$. *Then, for any problem instance* $\mathcal{P}$ *with the turret's angular velocity* $\omega \geq \frac{\theta}{D}$, *where* $D$ *is defined in equation 5.3, Algorithm Split is* $\frac{3N-1}{3\lfloor 0.5N \rfloor + 2}$*-competitive if* $v \leq \min\{\frac{1-\rho}{3D}, \frac{r_t-\rho}{2D}\}$.

*Proof.* First observe that although the turret can capture intruders from one half of the environment, the vehicle only captures at most two intervals out of all intervals that are in $W^k_{\text{Veh}}$ (the total number of intervals in $W^k_{\text{Veh}}$ will be determined shortly). Thus, in the worst-case, the intruders are released in the environment such that there are as many intruders possible in the vehicle's dominance region. Since $W^k_{\text{Veh}}$ is selected based on $W^k_{\text{Tur}}$, there cannot be more number of intruders in the vehicle's dominance region as than those in the turret's dominance region. This implies that there are equal number of intruders in each dominance region in every epoch in the worst-case. We now characterize the total number of intervals in the vehicle's dominance region.

If $N$ is even then, the vehicle's dominance region contains $\frac{N}{2}$ sectors and $3\frac{N}{2}$ intervals due to the three intervals of length $Dv$ each. Otherwise, the total number of intervals in the vehicle's dominance region is $3\lceil \frac{N}{2} \rceil$. The explanation is as follows. Observe that, for odd

$N$, the sector in the middle is contained in the turret's as well as the vehicle's dominance region. As the portion of the middle sector which is contained in the vehicle's dominance region may contain intruders and from the fact that the number of intervals must be an integer, we obtain that there are $3\lceil\frac{N}{2}\rceil$ intervals in the vehicle's dominance region. Since the total number of intervals in the environment is $3N$, this implies that the turret's dominance region has $3N - 3\lceil\frac{N}{2}\rceil = 3\lfloor\frac{N}{2}\rfloor$ intervals and not $3\lceil\frac{N}{2}\rceil$ intervals as we already accounted for the portion of the middle sector contained in the turret's dominance region in the vehicle's dominance region by using the ceil function. Intuitively, this means that there is no benefit for the adversary to release intruders in the portion of the middle sector contained in the turret's dominance region as the turret captures all intruders in its dominance region in an epoch. Thus, the adversary can have all intruders in a single interval within the turret's dominance region and the number of intruders that the turret capture remain the same, which is not the case in the vehicle's dominance region. We now account for the number of intruders jointly captured by the defenders in any epoch $k$.

Since at the start of every epoch $k$, the turret selects a dominance region based on the number of intruders on either side of the turret, it follows that the turret captures at least half of the total number of intruders that arrive in epoch $k$. This means that the turret captures intruders in all $3\lfloor\frac{N}{2}\rfloor$ intervals. The number of intruders captured by the vehicle in an epoch $k$ is determined as follows. Recall that in Algorithm Split, the vehicle's motion is independent of the turret's motion. The only information exchange that is required is the dominance region selected by the turret at the start of each epoch, which governs the number of sectors that the vehicle must account intruders in. Hence, this part of the analysis of accounting the number of intruders captured by the vehicle is identical to the proof of Lemma IV.5 in [10], so we only give an outline of the proof. From the fact that the vehicle's dominance region can have at most $3\lceil\frac{N}{2}\rceil$ intervals and by following similar steps as in proof of Lemma IV.5 from [10], it follows that for every two consecutively captured intervals, the vehicle loses at most $3\lceil 0.5N\rceil - 3$ intervals. Further, from Lemma 54 and by following similar steps as in the proof of Lemma IV.6 from [10], it follows that every lost interval is accounted for by the captured intervals of the turret and the vehicle. Thus, we obtain that the turret and the vehicle jointly capture at least $2 + 3\lfloor 0.5N \rfloor$ intervals of intruders and lose at most $3\lceil 0.5N \rceil - 3$ intruders in every epoch of Algorithm Split. Therefore, by assuming that there exists an optimal offline algorithm that can capture all $2+3\lfloor 0.5N \rfloor + 3\lceil 0.5N \rceil - 3 = 3N - 1$ intruder intervals in every epoch establishes that Algorithm Split is $\frac{3N-1}{3\lfloor 0.5N \rfloor + 2}$-competitive. This concludes the proof. □ □

Recall that the motion of the vehicle in Algorithm Split builds upon Algorithm SNP designed in [10], which was shown to be $\frac{3N-1}{2}$-competitive. A major drawback of Algorithm SNP was that its competitive ratio increases linearly with the number of sectors $N$. The following remark highlights that Algorithm Split does not suffer from this drawback and is effective under the same parameter regime as Algorithm SNP.

**Remark 11** (Heterogeneity improves competitive ratio of Algorithm Split)**.** *The competitive ratio of Algorithm Split is at most 2, achieved when $N \to \infty$. Further, for $r_t = 1$, the parameter regime that required by Algorithm Split ($v \leq \frac{1-\rho}{3D}$)) is the same as that of Algorithm SNP in [10].*

Further note that if $N$ is odd and $N \neq 1$, then the competitive ratio of Algorithm Split is higher than that for $N + 1$. This is because when $N$ is odd, the adversary can exploit the fact that there are higher number of intervals that the vehicle can lose as compared to that in the turret's dominance region. Finally, for $N = 2$, Algorithm Split is 1-competitive. The explanation is as follows. For $N = 2$, the two sectors of the environment overlap the two dominance region. Thus, in this case, the turret captures all intruders in one dominance region while the vehicle remains stationary at the resting point of the second dominance region, ensuring that all intruders that are released in the environment are captured.

Given that the turret can only move clockwise or anti-clockwise and from the requirement that the defenders must start their motions at the same time instant, the parameter regime of

**Algorithm 11:** Partition and Capture Algorithm

Turret's heading angle is $\frac{\theta}{3}$.
Vehicle is located at $(z_v, -\frac{\theta}{3})$.
**for** *each epoch* $k \geq 1$ **do**
    Compute $|V^k_{\text{left}}|, |T^k_{\text{right}}|$, and $|I^k|$.
    **if** $|I^k| \geq V^k_{\text{left}}$ *and* $|I^k| \geq T^k_{\text{right}}$ *holds* **then**
        **if** $|V^k_{\text{left}}| \geq |T^k_{\text{right}}|$ **then**
            Assign $V^k_{\text{left}}$ to the vehicle and $I^k$ to the turret.
        **else**
            Assign $T^k_{\text{right}}$ to the turret and $I^k$ to the vehicle.
        **end**
    **else**
        **if** $|V^k_{\text{left}}| < |I^k|$ *(resp.* $|T^k_{\text{right}}| < |I^k|$*)* **then**
            Assign $I^k$ to the vehicle (resp. turret).
        **else**
            Assign $V^k_{\text{left}}$ (resp. $T^k_{\text{right}}$) to the vehicle (resp. turret).
        **end**
    **end**
    Turn the defenders in an angular motion to the respective endpoint of the assigned set.
    Turn the defenders back to the initial position.
**end**

Algorithm Split is primarily defined by the time taken by the turret to sweep its dominance region. This means that by reducing the time taken by the turret to complete its motion, it is possible to achieve an algorithm with higher parameter regime. This is exploited in our next algorithm which is provably 1.5-competitive.

### 5.3.4 Partition and Capture (Part)

Algorithm Part, formally defined in Algorithm 11, partitions the environment into three equal dominance region, each of angle $\frac{2\theta}{3}$. We denote these dominance regions as $W_1$, $W_2$, and $W_3$, where $W_1$ denotes the leftmost dominance region. The idea is to move the vehicle and the turret similar to the motion of the turret in Algorithm Split and capture all intruders from two out of the three total dominance region in each epoch. The dominance region are determined as follows.

At the start of every epoch $k$, the turret's heading angle, measured from the $y$-axis, is

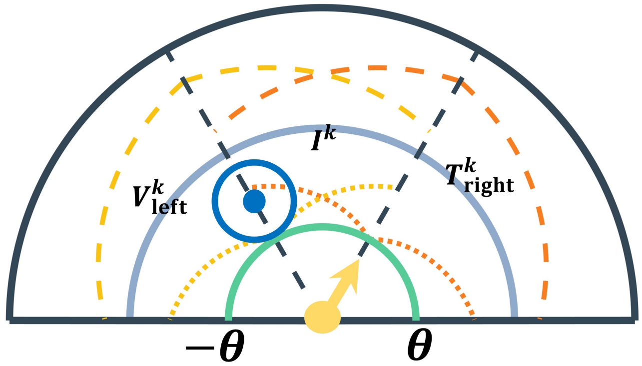



Figure 5.5 Description of Algorithm Part. The black dashed line denotes the partitioning of the environment, each of angle $\frac{2\theta}{3}$. The region between the orange (resp. yellow) dashed curve and the orange (resp. yellow) dot curve on the left (resp. right) side of the vehicle (resp. turret) denotes the $V_{\text{left}}$ (resp. $T_{\text{right}}$).

set to $\frac{\theta}{3}$. Similar to Algorithm Split, we describe two sets for the turret that characterize specific regions in the two dominance regions that surround the turret, i.e., $W_2$ and $W_3$. Intuitively, these sets corresponds to the locations in the environment that the turret can capture intruders at during its sweep motion.

$$T_{\text{right}}(\rho, v) := \{(y, \beta) : \rho + \frac{(\beta - \frac{\theta}{3})v}{\omega} \leq y \leq \min\{1, r_t + \frac{(\frac{5\theta}{3} - \beta)v}{\omega}\} \forall \beta \in [\frac{\theta}{3}, \theta]\},$$
$$T_{\text{left}}(\rho, v) := \{(y, \beta) : \rho - \frac{(\beta - \frac{\theta}{3})v}{\omega} \leq y \leq \min\{1, r_t + \frac{(\theta + \beta)v}{\omega}\} \forall \beta \in [-\frac{\theta}{3}, \frac{\theta}{3})\}.$$

Similarly, at the start of every epoch $k$, the vehicle is assumed to be located at $(z_v, -\frac{\theta}{3})$, where the angle is measured from the $y$-axis and $z_v$ is as defined for Algorithm SiR. Next, we define two sets that characterize a specific region in $W_1$ and $W_2$, respectively.

$$V_{\text{right}}(\rho, v) := \{(y, \beta) : \rho + (\tfrac{\theta}{3} + \beta) z_v v \leq y \leq \min\{1, z_v + r_c + (\theta - \beta) z_v v\} \ \forall \beta \in [-\frac{\theta}{3}, \frac{\theta}{3}]\},$$
$$V_{\text{left}}(\rho, v) := \{(y, \beta) : \rho - (\beta + \frac{\theta}{3}) z_v v \leq y \leq \min\{1, z_v + r_c + (\frac{5\theta}{3} + \beta) v z_v\} \ \forall \beta \in [-\theta, -\frac{\theta}{3})\}.$$

Let $T^k_{\text{right}}$, $T^k_{\text{left}}$,$V^k_{\text{right}}$, and $V^k_{\text{left}}$ denote the set of intruders contained in $T_{\text{right}}$, $T_{\text{left}}$, $V_{\text{right}}$, and $V_{\text{left}}$, respectively, at the start of an epoch $k$. Finally, denote $I^k$ as the set of intruders contained in $V^k_{\text{right}} \cap T^k_{\text{left}}$ (see Figure 5.5).

We now describe the motion of the defenders. The objective is to move the defenders such that intruders from any two sets out of $V^k_{\text{left}}$, $T^k_{\text{right}}$, and $I^k$ are captured. This requires assigning the defenders to the sets containing maximum number of intruders, which can be summarized into two cases.

**Case 1:** The set $I^k$ contains maximum number of intruders, i.e., $|I^k| \geq V^k_{\text{left}}$ and $|I^k| \geq T^k_{\text{right}}$ hold at the start of epoch $k$. This means that one of the defenders must be assigned to the set $I^k$. By determining which set has more intruders out of $V^k_{\text{left}}$ and $T^k_{\text{right}}$, Algorithm Part performs an assignment of the sets to the defenders. Mathematically, if $|I^k| \geq |V^k_{\text{left}}|$ and $|I^k| \geq |T^k_{\text{right}}|$, then

- If $|V^k_{\text{left}}| \geq |T^k_{\text{right}}|$, then the vehicle is assigned the set $V^k_{\text{left}}$ and the turret is assigned the set $I^k$.
- Otherwise, the vehicle is assigned the set $I^k$ and the turret is assigned the set $T^k_{\text{right}}$.

**Case 2:** $|V^k_{\text{left}}| < |I^k|$ or $|T^k_{\text{right}}| < |I^k|$ holds at the start of epoch $k$. This implies that at least one set out of $V^k_{\text{left}}$ and $T^k_{\text{right}}$ has the maximum number of intruders out of the three $V^k_{\text{left}}, I^k$, and $T^k_{\text{right}}$ sets. Then, the sets are assigned as follows:

- If $|V^k_{\text{left}}| < |I^k|$, then the vehicle is assigned the set $I^k$. Otherwise, the vehicle is assigned the set $V^k_{\text{left}}$.
- Similarly, if $|T^k_{\text{right}}| < |I^k|$, then the turret is assigned the set $I^k$. Otherwise, the turret is assigned the set $T^k_{\text{right}}$. Note that if the vehicle is already assigned set $I^k$ then that means that $|I^k| \geq |V^k_{\text{left}}|$ holds. Given the condition in Case 2, this implies that $|T^k_{\text{right}}| < |I^k|$ holds and the turret is assigned $T^k_{\text{right}}$. Thus, in Case 2, both defenders are never assigned the set $I^k$.

Once the sets are assigned, the vehicle turns as follows. If the set assigned to the vehicle is $I^k$, then the vehicle moves clockwise with unit speed in the direction perpendicular to its position vector until it reaches location $(z_v, \frac{\theta}{3})$. Upon reaching the location, the vehicle moves anti-clockwise with unit speed in the direction perpendicular to its position vector until it returns to location $(z_v, -\frac{\theta}{3})$. Otherwise (if the vehicle is assigned the set $V^k_{\text{left}}$), the vehicle moves anti-clockwise with unit speed in the direction perpendicular to its position vector until it reaches location $(z_v, -\theta)$. Upon reaching that location, the vehicle moves clockwise with unit speed in the direction perpendicular to its position vector until it returns to location $(z_v, -\frac{\theta}{3})$.

Before we describe the motion of the turret, we determine its angular speed to ensure that the defenders start an epoch at the same time instant. As we require that the defenders take the same amount of time to return to their starting locations in an epoch, we require that $\frac{4\theta}{3\omega} = \frac{4\theta}{3} z_v \Rightarrow \omega = \frac{1}{z_v}$, which means that the angular speed of the turret must be at least $\frac{1}{z_v}$.

We now describe the turret's motion in an epoch. Similar to the motion of the vehicle, if the set assigned to the turret is $I^k$, then the turret turns to angle $-\frac{\theta}{3}$ and then turns back to the initial heading angle $\frac{\theta}{3}$ with angular speed $\frac{1}{z_v}$. Otherwise, the turret turns to angle $\theta$ and then back to angle $\frac{\theta}{3}$ with angular speed $\frac{1}{z_v}$.

Analogous to Lemma 54, we have the following lemma which ensures that any intruder that was not considered for comparison at the start of epoch $k$ is considered for comparison at the start of epoch $(k+1)$.

**Lemma 56.** *Any intruder which lies beyond the sets $V^k_{\text{left}}, V^k_{\text{right}}, T^k_{\text{right}}$ and $T^k_{\text{left}}$ at the start of epoch $k$ will be contained in the sets $V^{k+1}_{\text{left}}, V^{k+1}_{\text{right}}, T^{k+1}_{\text{right}}$ and $T^{k+1}_{\text{left}}$, respectively, at the start of epoch $(k+1)$ if*

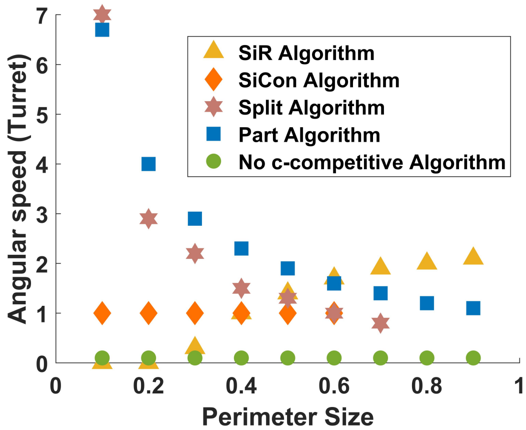
$$v \leq \min\left\{\frac{3(\min\{1, z_v + r_c\} - \rho)}{4\theta z_v}, \frac{3(r_t - \rho)}{4\theta z_v}\right\}$$

*Proof.* The proof is analogous to the proof of Lemma 54 and has been omitted for brevity.

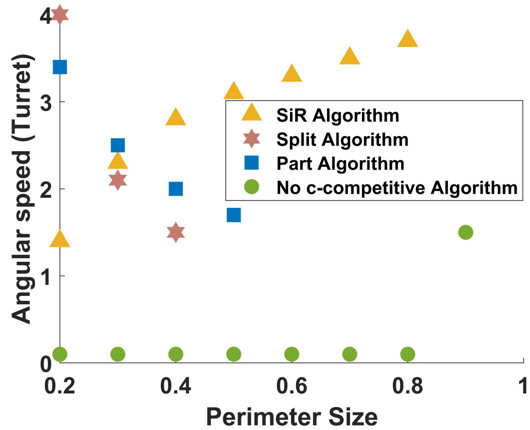
□ □

**Corollary 10.** *Any intruder that lies beyond the set $I^k$ at time instant $k_s$ will be contained in the set $I^{k+1}$ at the start of epoch $(k+1)$ if the conditions of Lemma 56 hold.*

*Proof.* The proof directly follows from the fact that Lemma 56 holds for both the defenders and $I^k$ represents the intersection of $V^k_{\text{right}}$ and $T^k_{\text{left}}$. □ □




(a) $(\omega, \rho)$ plot for $\theta = \frac{\pi}{4}, r_t = 1, r_c = 0.05$, and $v = 0.1$.




(b) $(\omega, \rho)$ plot for $\theta = \frac{\pi}{4}, r_t = 1, r_c = 0.1$, and $v = 0.3$.

Figure 5.6 $(\omega, \rho)$ plot for different values of $\theta, r_c, r_t$, and $v$. Markers represent a lower bound on the turret's angular speed.

**Theorem 57.** *Algorithm Part is* 1.5*-competitive for any problem instance* $\mathcal{P}$ *with* $\omega \geq \frac{1}{z_v}$ *that satisfies*

$$v \leq \min\left\{\frac{3(\min\{1, z_v + r_c\} - \rho)}{4\theta z_v}, \frac{3(r_t - \rho)}{4\theta z_v}\right\}$$

*Proof.* Observe that from Lemma 56 and Corollary 10, every intruder is accounted for and no intruder that is not considered for comparison in a particular epoch is lost under the condition on $v$. Now, from the definition of Algorithm Part, the defenders are assigned two sets out of the total three in every epoch. Further, the assignment is carried out in a way that the sets with maximum number of intruders are assigned to the defenders in every epoch. Assuming that there exists an optimal offline algorithm that captures all intruders from all three sets then, from Definition 1, competitive ratio of Algorithm Part is at most $\frac{3}{2}$. □ □

## 5.4 Numerical Observations

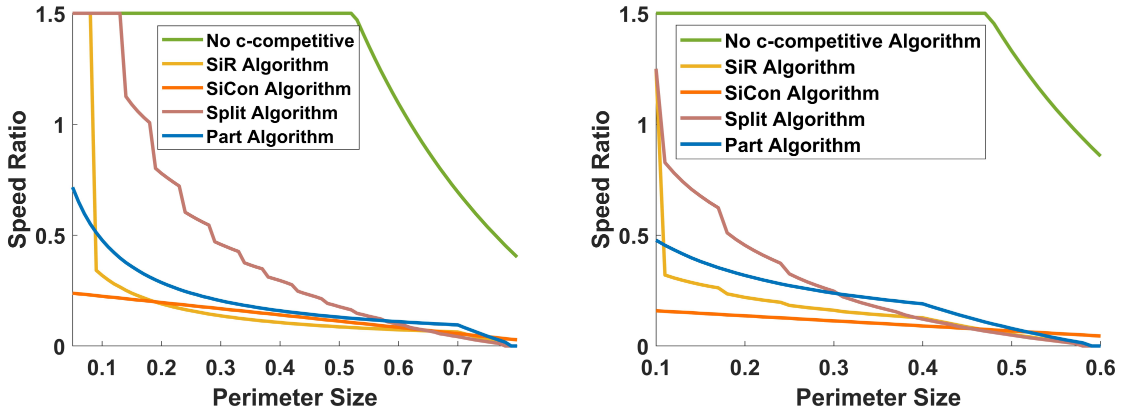
We now provide numerical visualization of the bounds derived in this work and emphasize on the $(v, \rho)$ and $(\omega, \rho)$ parameter regime. These plots allow the defenders to choose an appropriate online algorithm out of the four proposed, based on the values of the problem parameters.

### 5.4.1 $(\omega, \rho)$ Parameter Regime

Figure 5.6 shows the $(\omega, \rho)$ parameter regime plot for fixed values of $r_t, r_c, \theta$, and $v$ and provides insights into the requirement of the angular speed $\omega$ for different values of the $\rho$. Note that the markers represent a lower bound on the angular speed of the turret.

In Figure 5.6a, the condition in Theorem 50 for the existence of $c$-competitive algorithms is represented by the green circles. For all values of $0.1 \leq \rho \leq 0.9$, as the green circles are at $\omega = 0.1$, it implies that there exists a $c$-competitive algorithm for all values of $\omega \geq 0.1$ and the values of $v, r_c, r_t$, and $\theta$ selected for this figure. We now provide insights into the requirement on $\omega$ for our algorithms. Algorithm SiR, represented by the yellow triangle, requires higher angular speed for the turret as the radius of the perimeter increases. However, Algorithm Split and Algorithm Part, represented by the red star and blue square respectively, require lower angular speed for the turret when the radius of the perimeter is sufficiently large. Although counter intuitive, this can be explained as follows. Recall that Algorithm Part and Algorithm Split require the two defenders to be synchronous and the vehicle moves with a fixed unit speed. As the perimeter size increases, the time taken by the vehicle to complete its motion increases, which in turn requires lower values of $\omega$ to ensure synchronicity. Observe that for $\rho \geq 0.8$, there are no markers for Algorithm Split. This is because for $\rho \geq 0.8$ and the values of $\theta, r_t, r_c$ and $v$ considered for this figure, the condition defined for Algorithm Split in Theorem 55 is not satisfied for any $0 < \omega \leq 7$, implying that Algorithm Split is not $c$-competitive. Analogous conclusions can be drawn for Algorithms SiCon (resp. Part), represented by orange diamond (resp. blue square), for values of $\rho \geq 0.6$ (resp. $\rho \geq 0.9$). Finally, note that Algorithm SiCon requires $\omega \geq 1$ for all values of $\rho \leq 0.6$. This is because in this algorithm, the turret is required to move with unit angular speed to maintain synchronicity with the vehicle.

Analogous observations can be drawn in Figure 5.6b. For instance, when $\rho = 0.9$ and $\omega \geq 1.5$, there always exists a $c$-competitive algorithm with a finite constant $c$. Equivalently, there does not exist a $c$-competitive algorithm for $\omega < 1.5$, $\rho = 0.9$ and for the values of $r_t, r_c, \theta$,




(a) $(v, \rho)$ plot for $\theta = \frac{\pi}{3}, r_t = 0.8$, and $r_c = 0.05$.

(b) $(v, \rho)$ plot for $\theta = \frac{\pi}{2}, r_t = 0.6$, and $r_c = 0.1$.

Figure 5.7 $(v, \rho)$ plot for different values of $\theta, r_c$ and $r_t$.

and $v$ selected. Similarly, as $\rho$ increases, Algorithm SiR requires a faster turret whereas Algorithm Part and Algorithm Split can work with a slower turret to ensure synchronicity. Note that Algorithm Part and Split do not have markers beyond $\rho = 0.6$ and $\rho = 0.5$, respectively, which is lower than the values of $\rho$ in Figure 5.6a. This implies that, although the values of $r_c$ are slightly higher than those in Figure 5.6a, it is more difficult to capture intruders given the higher value of $v$. Finally, there are no markers for Algorithm SiCon as it is not $c$-competitive for the values of parameters selected for this figure.

### 5.4.2 $(v, \rho)$ Parameter Regime

Figure 5.7 shows the $(v, \rho)$ parameter regime for fixed values of $\theta, r_c$, and $r_t$. Since Algorithms Split, Part, SiCon require fixed but different values of $\omega$, we set $\omega = \max\{\frac{1}{\min\{1, r_t + r_c\}}, \frac{1}{z_v}, \frac{\theta}{D}\}$. Note that the value of $\omega$ for this figure depends on the value of $\rho$ as $z_v$ is a function of $\rho$.

Figure 5.7a shows the $(v, \rho)$ parameter regime plot with $\theta, r_t$, and $r_c$ set to $\frac{\pi}{3}, 0.8$, and 0.05, respectively. For any value of parameters $\rho$ and $v$, for instance 0.7 and 1, respectively, that lie beyond the green curve, there does not exists a $c$-competitive algorithm. For any value of parameters $\rho$ and $v$ that lie below the yellow curve, Algorithm SiR is 1-competitive. Similarly, for any value of parameters $\rho$ and $v$ that lie below the blue curve, Algorithm Part is 1.5-competitive. Analogous observations can be made for Algorithm Sicon and Algorithm

Split. Note that for parameter regime that lies below the yellow curve, Algorithm Part is not effective as there exists Algorithm SiR with a better competitive ratio. For instance, for $\rho = 0.2$ and $v = 0.2$, it is better to use Algorithm SiR as it has a lower competitive ratio. Observe that for very high values of $\rho$, Algorithm SiCon is the most effective as it has the highest parameter regime curve. Finally, the light red curve of Algorithm Split is divided into *regions* where each region corresponds to a specific competitiveness. An important characteristic for Algorithm Split is that it can be used to determine the tradeoff between the competitiveness and the desired parameter regime for a specific problem instance.

Figure 5.7b shows the $(v, \rho)$ parameter regime plot with $\theta, r_t$, and $r_c$ set to $\frac{\pi}{2}, 0.6$, and 0.1, respectively. Note that the green curve, which represents the curve for Theorem 50, is shifted slightly upwards as compared to in Figure 5.7a. This follows from the two cases considered in the proof which is based on the capture capability of the defenders (vehicle is now more capable and Theorem 50 is independent of $r_t$). As the angle of the environment increases and the engagement range of the turret decreases, it is harder to capture intruders. This is visualized in Figure 5.7b as the curves for all the algorithms have shifted downward compared to those in Figure 5.7a. Finally, for values of $\rho > 0.3$, Algorithm Part is more effective than Algorithm Split only if the competitive ratio of Algorithm Split is less than 1.5 for the chosen values of parameters. Similar to the curve of Algorithm Split, note that the curve for Algorithm SiR is also divided into regions. This is because of the different values of $\omega$ for different perimeter sizes.

## 5.5 Discussion

In this section, we provide a brief discussion on the time complexity of our algorithms and how this work extends to different models of the vehicle. We start with the time complexity of our algorithms.

### 5.5.1 Time Complexity

We now establish the time complexity of each our algorithms and show that they can be implemented in real time if the information about the total number of intruders in every epoch is provided to the defenders.

Algorithm SiR and SiCon : Since Algorithm SiR and SiCon are open loop algorithms, the time complexity is $O(1)$.

Algorithm Split : There are three quantities that must be computed at the start of every epoch of Algorithm Split, i.e., $|P^k_{\text{right}}|, |P^k_{\text{left}}|$, and $N_{p^*}$. Since $N_{p^*}$ is determined using a max() function over $N$ sectors, its time complexity is $O(N)$. Similarly, determining the sets $P^k_{\text{right}}$ and $P^k_{\text{left}}$ also have a time complexity of $O(n)$, where $n$ is the number of intruders in an epoch. This yields that the time complexity of Algorithm Split is $O(\max\{n, N\})$. Recall that $N$ is finite as $r_c > 0$. Thus, in the case when $n \to \infty$, if the information about the number of intruders in $P^k_{\text{right}}$ and $P^k_{\text{left}}$ is provided to the defenders (through some external sensors), then this algorithm can be implemented in real time.

Algorithm Part : Similar to Algorithm Split, Algorithm Part computes $|T^k_{\text{right}}|, |V^k_{\text{left}}|$, and $|I^k|$ at the start of every epoch which yields that the time complexity of Algorithm Part is $O(n)$. This requires that the information about the total number of intruders in each of these sets must be provided to the defenders, for $n \to \infty$, to implement this algorithm in real time.

We now discuss how this work extends to different motion model of the vehicle.

### 5.5.2 Different Motion Models for the Vehicle

Observe that the analysis in this work is based upon two quantities; first, the time taken by the intruders to reach the perimeter and second, the time taken by the defenders to complete the motion. This work can be extended to other models for the vehicle, for instance double integrator, by suitably modifying the time taken by the vehicle to complete its motion. By doing so, it may be that the parameter regimes may be lower than in Figure 5.7 but the bounds on the competitive ratios will remain the same. The reason that the parameter regimes will be lower is as follows. Note that the parameter regimes are characterized by the conditions determined for each of the algorithms. Essentially, these conditions are determined by requiring the intruders to be sufficiently slow such that they take more time to reach the perimeter than the time taken by the vehicle to complete its motion. For a different model of the vehicle, such as the Dubins model, the path and the time taken by the vehicle to complete its motion can be determined by suitably incorporating the turn radius. Precise dependence of the competitiveness of such realistic models will be a topic of a future investigation.

## 5.6 Conclusions

This work analyzed a perimeter defense problem in which two cooperative heterogeneous defenders, a mobile vehicle with finite capture range and a turret with finite engagement range, are tasked to defend a perimeter against mobile intruders that arrive in the environment. Our approach was based on a competitive analysis that first yielded a fundamental limit on the problem parameters for finite competitiveness of any online algorithm. We then designed and analyzed four algorithms and established sufficient conditions that guaranteed a finite competitive ratio for each algorithm under specific parameter regimes.

In this chapter, we considered a turret whose motion is not coupled with the vehicle. In the next chapter, we consider a scenario in which a turret is attached to a mobile defender.

# CHAPTER 6

## OPTIMAL CONTROL OF A DUBINS-LASER SYSTEM

In the previous chapter, we considered a team of heterogeneous defenders whose motion was independent of each other. However, in some scenarios, it is possible that the motion of one defender depends on the motion of another defender. Examples of such scenarios include a UAV with a gimbaled camera having an ability to rotate clockwise or anti-clockwise. Note that the angular displacement of the camera is affected differently when the UAV turns in the same or the opposite direction. This chapter delves into systems of this nature.

This chapter considers an optimal control problem in the plane for a *Dubins-Laser System* which consists of a Dubins vehicle, that moves with unit speed, and a laser which is attached to the vehicle. The laser has a finite range and can rotate either clockwise or anti-clockwise with a bounded angular speed. The environment consists of a single static target. The objective of this work is to determine a time optimal control strategy for the Dubins-Laser System such that the following two conditions jointly hold at final time.

- The target is within the range of the laser and
- the laser is oriented towards the target.

The contributions of this chapter is as follows:

1. **Optimal Control of Dubins-Laser System:** We formulate a novel optimal control problem of a Dubins vehicle with a gimbaled laser system. The laser, having a finite range, is modelled as a single integrator and rotates either clockwise and anti-clockwise in the environment. The environment consists of a single static target. The aim is to determine an optimal control strategy of the Dubins-Laser System such that the target is within the range of the laser and the laser is oriented towards the target in minimum time.

2. **Set of Candidate Paths:** We establish multiple, including cooperative, properties of the Dubins-Laser System. We also determine the set of candidate paths for the same. In particular, we establish that the set of candidate paths consists of 13 paths, i.e., the shortest path for the Dubins-Laser System is of type $\{CSC, CC\}$ or a subsegment of it.

3. **Semi-Analytical Solution of the Optimal Path:** Finally, we establish that the optimal path can be efficiently determined by solving a set of non-linear polynomial equations, thus, providing a semi-analytical solution to the problem.

This chapter is organized as follows. In Section 6.1, we provide relevant background information and in Section 6.2, we formally define the problem. Section 6.3 characterizes the optimal control for the Dubins-Laser system, Section 6.4 characterizes multiple properties of the shortest path, and Section 6.5 characterizes the solution of the shortest path. In Section 6.6, we provide numerous numerical results and finally, Section 6.7 summarizes this chapter.

## 6.1 Preliminaries

In this section, we provide some relevant background information that will be instrumental in understanding the content of this chapter.

### 6.1.1 The Minimum Principle

For a given a positive integer $n$, let $\mathbf{x}(t) = \begin{bmatrix} x_1(t), \ldots, x_n(t) \end{bmatrix}$ denote the state of a dynamical system given by

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t)),$$

where $\mathbf{u}(t) : \mathbb{R} \to \mathbb{R}^n$ is a control signal and $\mathbf{u} \in U$ where $U$ denotes the set of admissible control signals. Then, given the boundary conditions (initial and/or final state), a minimum time optimal control problem is to find controls $\mathbf{u}^*(t) \in U$ such that the cost $J(\mathbf{u}(t)) := \int_0^{t_f} dt = t_f$ is minimized, where $t_f$ denotes the final time. This means that we want to minimize the time to go from the initial to the final state.

Let $p_0, \ldots, p_n$ be the dual variables, also known as the *costates*, and $\mathbf{p} = [p_0, \ldots, p_n] : \mathbb{R} \to \mathbb{R}^{n+1}$ be the adjoint vector. Then, for a given control $\mathbf{u}(t)$ and an initial condition, a *Hamiltonian* is defined by $H(\mathbf{p}(t), \mathbf{x}(t), \mathbf{u}(t)) : \mathbb{R}^{3n+1} \to \mathbb{R}$

$$H(\mathbf{p}(t), \mathbf{x}(t), \mathbf{u}(t)) = \sum_{j=0}^{n} p_j f_j(\mathbf{x}(t), \mathbf{u}(t)),$$

where $f_0 = 1$ and $f_j(\mathbf{x}(t), \mathbf{u}(t))$, for $j \geq 1$, denotes the $j$th element of $f(\mathbf{x}(t), \mathbf{u}(t))$. The costates are a solution of the *adjoint* system

$$\dot{p_j} = -\sum_{i=0}^{n} \frac{\partial f_j}{\partial x_i}(\mathbf{x}(t), \mathbf{u}(t)), \quad j \in \{0, \ldots, n\}.$$

Then, the minimum principle states the following.

**Theorem 58** (Minimum Principle [61])**.** *If $\boldsymbol{u}^*(t)$ is an admissible optimal control, then there exists a nonzero adjoint vector $\boldsymbol{p}(t)$ such that*

1. $H(\boldsymbol{p}(t), \boldsymbol{x}(t), \boldsymbol{u}^*(t)) = \min_{u \in U} H(\boldsymbol{p}(t), \boldsymbol{x}(t), \boldsymbol{u}(t))$, $\forall t \in [0, t_f]$.
2. $H(\boldsymbol{p}(t), \boldsymbol{x}(t), \boldsymbol{u}^*(t)) = 0$ *and* $p_0 \geq 0$ $\forall t \in [0, t_f]$.

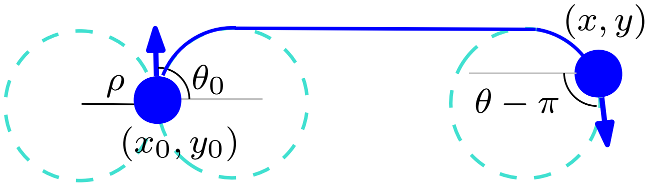
### 6.1.2 Optimal Control of a Dubins vehicle

A Dubins vehicle is a vehicle that moves with a constant speed (usually normalized to unity) and has a minimum turn constraint. Formally, a Dubins vehicle has the following kinematic model:

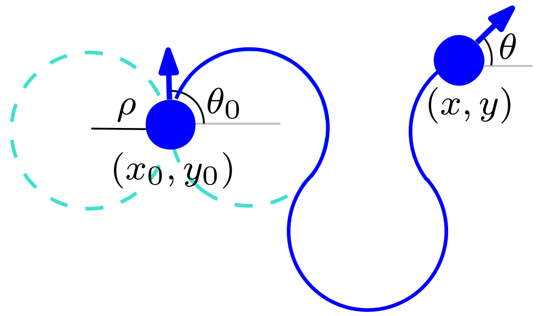
$$\dot{x} = \cos(\theta), \quad \dot{y} = \sin(\theta), \quad \dot{\theta} = \frac{u}{\rho},$$

where $(x, y) \in \mathbb{R}^2$ denotes the position of the Dubins vehicle in the plane, $\theta \in \mathbb{S}^1$ denotes the orientation of the vehicle, $\rho > 0$ denotes the minimum turn radius, and $u \in [-1, 1]$ denotes the control.

The classical problem of finding the shortest path for a Dubins vehicle from a given initial state to a given final state was studied by Dubins in [27]. Dubins, through geometry, established that for given endpoints and orientation of the vehicle at these endpoints, the




(a) An illustration of a $RSR$ type path.




(b) An illustration of a $RLR$ type path.

Figure 6.1 Illustration of a $CSC$ and a $CCC$ type path for a given initial state $[x_0, y_0, \theta_0]$ and final state $[x, y, \theta]$. The blue dot represents the Dubins vehicle and the blue arrow represents its orientation. The light blue circles denote the circle of radius $\rho$ subtended by the circular arcs.

shortest path is one out of 15 candidates paths. He further established that these candidate paths consists of only straight line segments and circular arcs. Later, authors in [15] established the same result by applying the minimum principle (also known as the Pontryagin's maximum principle). We now describe the set of candidate paths for the Dubins vehicle determined in [15] and [27]. We will first describe the classical result for the Dubins vehicle and then provide a brief explanation about the set of candidate paths.

**Theorem 59** (Shortest path for a Dubins vehicle [15])**.** *Any shortest path in the plane between given endpoints and orientations is either of type $CSC$ or of type $CCC$, or a subsegment of it, where $C$ denotes a circular arc of radius $\rho$ and $S$ denotes a straight line segment.*

The aforementioned result states that the shortest path for a Dubins vehicle is formed of a sequence of at most three segments, joined at isolated points called as *inflexion points*. Further, the shortest path can have at most one straight line segment which implies that the shortest path is of type $CSC$ or $CCC$ or a subsegment of it. The subsegments of a path of type $CSC$ are $CS, SC, S$. Similarly, the subsegments of a path of type $CCC$ are $CC, C$. Thus, the set of candidate paths is $\{CSC, CS, SC, S, CCC, CC, C\}$ or compactly denoted as $\{CSC, CCC\}$. Note that a circular arc can be a clockwise (or right) turn or an anticlockwise (or left) turn. A clockwise turn is denoted as $R$ and an anticlockwise turn is denoted as $L$.

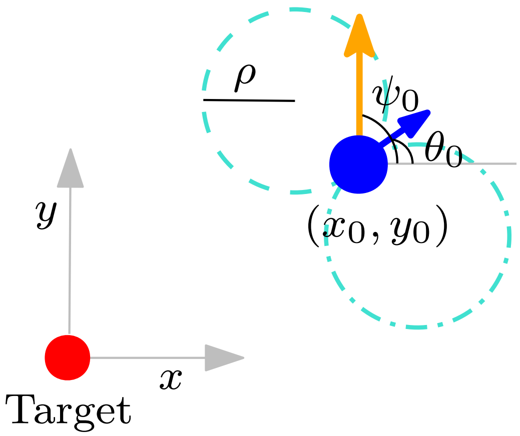



Figure 6.2 Problem Description. The red dot depicts the static target located at the origin and the blue dot represents the Dubins vehicle. The laser is depicted by the yellow (longer) arrow and the blue (shorter) arrow depicts the orientation of the Dubins vehicle.

Considering all possible combinations of $R$ and $L$ turns yield the total number of candidate paths. For instance, a $CSC$ path can be any one out of the set $\{RSR, RSL, LSL, LSR\}$. Figure 6.1a and Figure 6.1b illustrates a $CSC$ and a $CCC$ type path, respectively.

When the orientation of the Dubins vehicle is not specified at the final location, then the optimal control problem of determining the shortest path for the Dubins vehicle for given endpoints and orientation at the initial location is known as a Relaxed Dubins problem. In this case, the set of candidate paths reduces to $\{CS, CC\}$ [19].

### 6.2 Problem Description

We consider an optimal control problem of a Dubins-Laser (Dub-L) System in the $XY$ plane that consists of a Dubins vehicle [27] and a laser which is attached to the vehicle (Fig. 6.2). The laser has a finite range $R > 0$. The vehicle, with a minimum turn radius $\rho > 0$, moves with a constant unit speed and the laser, modelled as a single integrator, has the ability to rotate either clockwise or anti-clockwise with speed $\omega \leq \omega_M$, where $\omega_M > 0$. The environment consists of a static target, assumed to be located at $(x_T = 0, y_T = 0)$. The state vector of the Dub-L system is denoted by $\mathbf{x} = \begin{bmatrix} x & y & \theta & \psi \end{bmatrix}^\top$, where $(x, y) \in \mathbb{R}^2, \theta \in \mathbb{S}^1$, and $\psi \in \mathbb{S}^1$ denotes the position of the vehicle, the heading direction of the vehicle measured in the anticlockwise direction from the positive $X$-axis, and the orientation of the laser measured in the anticlockwise direction from the positive $x$-axis, respectively. The kinematic equations that describe the motion of the Dub-L System is

$$\dot{x} = \cos(\theta), \dot{y} = \sin(\theta), \dot{\theta} = \frac{u}{\rho}, \dot{\psi} = \frac{u}{\rho} + \omega,$$

where $u \in [-1, 1]$ denotes the control input of the Dubins vehicle, and $\omega \in [-\omega_M, \omega_M]$ denotes the angular speed of the laser. We denote the control vector as $\mathbf{u} = \begin{bmatrix} u & \omega \end{bmatrix}^\top$ and is defined as $\mathbf{u} : [0, \infty) \to \mathbb{R}^2$.

The target is said to be *captured* at time $t$ if the distance between the target and the Dub-L system is at most $R$ and the laser is oriented towards the target. Mathematically, a target is captured at time $t$ if the following two conditions jointly hold:

$$\begin{aligned} x^2(t) + y^2(t) &\leq R^2, \\ \pi + \arctan\left(\frac{y_T - y(t)}{x_T - x(t)}\right) - \psi(t) &= 0. \end{aligned} \tag{6.1}$$

Given an initial state $\mathbf{x}(0) := \mathbf{x}_0 = \begin{bmatrix} x_0 & y_0 & \theta_0 & \psi_0 \end{bmatrix}^\top$ of the Dub-L System, the objective of this work is to determine an optimal trajectory for the Dub-L System such that the target is captured in minimum time, i.e., equations (6.1) hold at final time $t_f$.

The proposed optimal control problem may have non-unique solutions for some initial conditions and problem parameters. For instance, let $x^*(t), y^*(t), \theta^*(t)$ be the time optimal trajectory for the Dub-L system for all $t \in [0, t_f]$, where $t_f$ denotes the final time. Further, suppose that the initial condition is such that the optimal control for the laser is

$$\omega^*(t) = \begin{cases} \omega_M, & \text{if } t \in [0, t']. \\ 0, & \text{otherwise} \end{cases}$$

for some time $t' < t_f$. Then, for some very small $\epsilon > 0$, there exists another optimal control for the laser

$$\tilde{\omega}^*(t) = \begin{cases} \omega_M, & \text{if } t \in [\epsilon, t' + \epsilon]. \\ 0, & \text{otherwise} \end{cases}$$

for which the final time at which the target is captured remains the same. Thus, to characterize one trajectory of the laser out of infinitely many, we make the following assumption.

**Assumption 1.** *At time instant 0, the laser is switched off and is switched on at time $0 \leq t_l \leq t_f$. Once the laser is switched on, it remains switched on until the target is captured.*

The time instant $t_l$ is such that the laser remains switched off for maximum amount of time. To ensure that the target is within the range of the laser, the laser must be switched on at some time instant $0 \leq t_l \leq t_f$. Hence, the time instant at which the laser must be switched on becomes an optimization variable that needs to be determined to characterize the trajectory of the Dub-L System.

Let $i(t)$ denote whether the laser is switched on or off. Formally,

$$i(t) = \begin{cases} 0, & \text{if } t < t_l, \\ 1, & \text{otherwise.} \end{cases}$$

Then, based on Assumption 1, we model this problem as a two stage optimal control problem with the following kinematic model.

$$\dot{\mathbf{x}} = \sum_{k=1}^{2} i(t) f_k(\mathbf{x}, \mathbf{u}),$$

where

$$f_1(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & \frac{u}{\rho} & \frac{u}{\rho} \end{bmatrix}^\top$$
$$f_2(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & \frac{u}{\rho} & \frac{u}{\rho} + \omega \end{bmatrix}^\top.$$

We now formally describe the objective of this work.

**Problem Statement:** Given an initial state vector $\mathbf{x}_0$, the aim is to determine an optimal control $\mathbf{u}^* = \begin{bmatrix} u^* & \omega^* \end{bmatrix}^\top$ and time instant $t_l^* \geq 0$, such that equation (6.1) holds in minimum time.

We now apply the necessary conditions and characterize a cooperative nature between the Dubins vehicle and the laser in the next section.

### 6.3 Necessary Conditions

Let $\mathbf{p} = \begin{bmatrix} p_x & p_y & p_\theta & p_\psi \end{bmatrix} \in \mathbb{R}^4$ denote the adjoint vector corresponding to $x, y, \theta$, and $\psi$, respectively. Then, given that the aim is to minimize the time before and after time $t_l$, we can write the Hamiltonian $H_k, \forall k \in \{1, 2\}$, as

$$H_k(\mathbf{x}, \mathbf{u}, \mathbf{p}, p_0) = p_0 + \mathbf{p} \cdot f_k(\mathbf{x}, \mathbf{u}),$$

where $p_0 \geq 0$ is the abnormal multiplier. Although the costates may be discontinuous at time $t_l$, they must satisfy the adjoint equations in the time interval $t \in [t_0, t_f], t \neq t_l$ which yields

$$\dot{p}_x(t) = \dot{p}_y(t) = \dot{p}_\psi(t) = 0,$$
$$\dot{p}_\theta(t) = p_x(t)\sin(\theta) - p_y(t)\cos(\theta).$$

This implies that $p_x(t), p_y(t)$, and $p_\psi(t)$ are constant in the time interval $[t_0, t_l)$ and $(t_l, t_f]$. Let $t_l^-$ (resp. $t_l^+$) denote the time just before (resp. after) time $t_l$. Then, at time $t_l$, since there is no cost associated to switching from $f_1(\mathbf{x}, \mathbf{u})$ to $f_2(\mathbf{x}, \mathbf{u})$ and the state does not exhibit jumps at time instant $t_l$, we obtain $p_x(t_l^-) = p_x(t_l^+)$, $p_y(t_l^-) = p_y(t_l^+)$, $p_\psi(t_l^-) = p_\psi(t_l^+)$, $p_\theta(t_l^-) = p_\theta(t_l^+)$, and $H_1(t_l^-) = H_2(t_l^+)$. Thus, integrating $p_x, p_y, p_\theta$, and $p_\psi$ yields

$$p_x(t) = c_x, p_y(t) = c_y, p_\psi(t) = c_\psi \text{ and}$$
$$p_\theta(t) = c_x y(t) - c_y x(t) + c_0, \forall t \in [t_0, t_f]$$

where $c_x, c_y, c_\psi$, and $c_0$ are some constants.

The following lemma will be useful in characterizing the singular control for the Dub-L System.

**Lemma 60.** *Suppose that the path of the Dub-L System consists of straight line segments. Then, $p_\theta + p_\psi = 0$ at the inflexion points and on the line segments.*

*Proof.* If $p_\theta + p_\psi = c_x y(t) + c_y x(t) = 0$ for any non-zero interval of time, then it follows that the path is a straight line in that interval. Finally, as $u$ changes sign at the inflexion point between two arcs and since $p_\theta + p_\psi$ is continuous, it follows that $p_\theta + p_\psi = 0$ at the inflexion point between two arcs. ☐

From the minimum principle [61] and Lemma 60, we obtain

$$\arg\min_u H_1 = \arg\min_u H_2 \Rightarrow u^* = \begin{cases} -1, p_\theta + p_\psi > 0 \\ +1, p_\theta + p_\psi < 0 \\ 0, p_\theta + p_\psi = 0, \end{cases} \tag{6.2}$$

$$\arg\min_\omega H_2 \Rightarrow \omega^* = \begin{cases} -\omega_M, p_\psi > 0 \\ \omega_M, p_\psi < 0. \end{cases} \tag{6.3}$$

where we used the fact that $p_\psi(t_l^-) = p_\psi(t_l^+)$, and $p_\theta(t_l^-) = p_\theta(t_l^+)$. From equation (6.2), it follows that the path of the Dub-L System comprises of arcs of radius $\rho$ (when $u = +1$ or $u = -1$) and straight line segments (when $u = 0$). Note that equation (6.3) does not characterize the control $\omega^*$ when $p_\psi = 0$. We characterize the control when $p_\psi = 0$ in Section 6.4.

Let $S$ denote a straight line segment and $C$ denote a circular arc of radius $\rho$. As a circular arc can be a right turn or a left turn, in the sequel, a right turn circular arc will be denoted as $R$ and a left turn circular arc will be denoted as $L$. The following two results follow immediately from Lemma 60.

**Corollary 11.** *On any optimal path, all inflexion points and all of the line segments are collinear with the target location.*

*Proof.* The proof is analogous to that of the proof of Lemma 7 of [18] and has been omitted for brevity. ☐

**Corollary 12.** *The segment of the optimal path before and after time instant $t_l$ is one out of $\{R|R, L|L, S|S\}$.*

*Proof.* The result follows from the fact that $p_\theta$ and $p_\psi$ is continuous at time instant $t_l$. ☐

The transversality conditions yield

$$\begin{aligned} p_x(t_f) &= \lambda_1 x(t_f) - \lambda_2 \frac{y(t_f)}{x^2(t_f)+y^2(t_f)} \\ p_y(t_f) &= \lambda_1 y(t_f) + \lambda_2 \frac{x(t_f)}{x^2(t_f)+y^2(t_f)} \\ p_\theta(t_f) &= 0, p_\psi(t_f) = -\lambda_2, \end{aligned} \tag{6.4}$$

where $\lambda_1$ and $\lambda_2$ are Lagrange multipliers. Since $p_x(t_f) = c_x$ and $p_y(t_f) = c_y$, by substituting $p_x(t_f)$ and $p_y(t_f)$ in the equation of $p_\theta(t)$ and using the fact that $p_\theta(t_f) = 0$ yields $\lambda_2 = c_0 \Rightarrow c_\psi = -c_0$. Thus, equation (6.2) and (6.3) becomes

$$u^*(t) = \begin{cases} -1, c_x y(t) - c_y x(t) > 0 \\ +1, c_x y(t) - c_y x(t) < 0 \\ 0, c_x y(t) - c_y x(t) = 0, \end{cases} \tag{6.5}$$

$$\omega^*(t) = \begin{cases} -\omega_M, c_0 < 0 \\ \omega_M, c_0 > 0. \end{cases} \tag{6.6}$$

We now characterize a cooperative nature between the laser and the Dubins vehicle.

**Lemma 61.** *Suppose that the optimal trajectory of the Dub-L system ends with a circular arc, i.e., the final segment of the optimal path is of type $C$. Then, $\omega^*(t)$ and $u^*(t)$ have the same sign in the final $C$ type segment.*

*Proof.* Consider that at time $t_f$, $u^*(t_f) = -1$ and $\omega^*(t_f) = \omega_M$. As $p_\theta(t_f) = 0$, it follows that $c_x y(t_f) - c_y x(t_f) = -c_0$. As $u^*(t_f) = -1 \Rightarrow c_x y(t_f) - c_y x(t_f) > 0$ which yields that $c_0 < 0$. This implies that $\omega^*(t_f) = -\omega_M$ which is a contradiction. Thus, $u^*$ and $\omega^*$ have the same sign at time instant $t_f$. Finally, since $c_\psi = -c_0$ is constant in $[t_0, t_f]$, and $u^*$ does not change sign in the final C segment, the result follows. The case when $u^* = 1$ and $\omega^* = -\omega_M$ is analogous and has been omitted for brevity. This concludes the proof. □

### 6.4 Characterization of Shortest Path

In this section, we characterize the shortest path of the trajectories. We start with the case when $p_\psi = 0$.

#### 6.4.1 Characterization of Shortest Path of Trajectories when $p_\psi = 0$

In this section, we establish that when $p_\psi = 0$, the problem is equivalent to solving the Relaxed Dubins problem to a circle. We start with a sufficient condition for $p_\psi = 0$.

**Lemma 62.** *The costate $p_\psi = 0$ if $t_l^* > t_0$.*

*Proof.* Since $t_l > t_0$ and there is no cost associated to switching from $f_1(\mathbf{x}, \mathbf{u})$ to $f_2(\mathbf{x}, \mathbf{u})$ at time instant $t_l$, it follows that $H_1(t_l^-) = H_2(t_l^+)$ which yields

$$\begin{aligned} &p_0 + p_x \cos\theta(t_l^-) + p_y \sin\theta(t_l^-) + \left(p_\theta(t_l^-) + p_\psi\right) \frac{u(t_l^-)}{\rho} = \\ &p_0 + p_x \cos\theta(t_l^+) + p_y \sin\theta(t_l^+) + p_\theta(t_l^+) \frac{u(t_l^+)}{\rho} + \\ &p_\psi \left( \frac{u(t_l^+)}{\rho} + \omega(t_l^+) \right) \Rightarrow p_\psi \omega(t_l^+) = 0, \end{aligned} \tag{6.7}$$

where we used $p_x(t_l^-) = p_x(t_l^+), p_y(t_l^-) = p_y(t_l^+), p_\psi(t_l^-) = p_\psi(t_l^+), p_\theta(t_l^-) = p_\theta(t_l^+)$. Since $\omega = 0$ is non-optimal, it follows that $p_\psi = 0$. This concludes the proof. □

The next result characterizes the set of candidate paths for the Dub-L System when $p_\psi = 0$.

**Lemma 63.** *If $p_\psi = 0$, then the optimal path of the Dub-L System must be of type $\{CS, CC\}$ or a subsegment of it.*

*Proof.* If $p_\psi = 0$, then Lemma 60, Corollary 11, and Corollary 12 still holds. Further, from transversality conditions, $\lambda_2 = 0$ meaning that the constraint that the laser must be oriented towards the target at final time is inactive. In other words, this means that the laser has sufficient time to orient itself towards the target while the Dubins vehicle moves towards the final location. Thus, it follows that when $p_\psi = 0$, the problem is equivalent to solving the Relaxed Dubins problem to a circle. Finally, analogous to Lemma 2.1 in [19] it can be shown that the $CSC$ and the $CCC$ path are not optimal for the Relaxed Dubins problem to a circle. Thus, the optimal path of the Dub-L System when $p_\psi = 0$ is of type $\{CS, CC\}$ or a subsegment of it. This concludes the proof. □

**Corollary 13.** *$x^2(t_f) + y^2(t_f) = R^2$ holds at time $t_f$ if $p_\psi = 0$.*

*Proof.* The result directly follows from Lemma 63 as the problem is equivalent to the Relaxed Dubins problem to a circle. □

We now determine the control $\omega$ when $p_\psi = 0$.

**Lemma 64.** *If $p_\psi = 0$, then*

$$\omega^* = \begin{cases} \omega_M, \ \textit{if}\ \psi(t_l^*) - \psi(t_f) <= \pi \\ -\omega_M, \ \textit{otherwise.} \end{cases}$$

*Proof.* The proof is straightforward and follows from the fact that $\lambda_2 = p_\psi = 0$ which implies that the laser has sufficient time to turn towards the final orientation while the vehicle moves to the final location. □

In this subsection, we characterized the shortest path for the Dub-L System when the speed of the laser is such that it has sufficient time to turn towards the final orientation. We now characterize trajectories when the laser must be switched on at time $t_0$.

### 6.4.2 Characterization of Shortest Path of Trajectories when $p_\psi \neq 0$

This subsection characterizes the main result of this work that the shortest path of the Dub-L System is of type $\{CSC, CC\}$ or a subsegment of it. As a consequence of Lemma 62, we consider that $t_l = t_0$ in this subsection. We start with the following remark.

**Remark 12.** *No path of type $\{CCCC\}$ taken by the Dub-L System can be optimal.*

The explanation is as follows. Suppose that at time $t_f$, the vehicle reaches location $(x(t_f), y(t_f))$ with an orientation $\theta(t_f)$. If $\omega_M = 0$, then from Lemma 11 in [18], it follows that a $\{CCCC\}$ path is not optimal. Let $L_{\omega_M=0}$ (resp. $L_{\omega_M>0}$) denote the length of the path when $\omega_M = 0$ (resp. $\omega_M > 0$). From Lemma 61, it follows that that $L_{\omega_M>0} < L_{\omega_M=0}$. This implies that a path of type $\{CCCC\}$ cannot be optimal in the interval $[t_l^*, t_f]$ for any $\omega_M > 0$.

Since a path of type $CCCC$ is not optimal, we turn our attention towards characterizing properties of path of type $CCC$ which will be instrumental is establishing the main result of this work.

**Lemma 65.** *Suppose that a path of type CCC for the Dub-L system exists and* $\omega_M \neq \frac{u}{\rho}$. *Then,* $0 < \Delta\psi_2 < \pi$ *for any* $\omega_M \neq \frac{1}{\rho}$, *where* $\Delta\psi_2$ *denotes the angle that the laser rotates in the middle* $C$ *segment of the* $CCC$ *path.*

*Proof.* Denote the path of type $CCC$ as $\tau$ and suppose that $\Delta\psi_2 \geq \pi$. Without loss of generality, we assume that the laser is turning clockwise in trajectory $\tau$. Then, the time taken by the turret to rotate in the middle $C$ segment is $T = \frac{\Delta\psi_2}{\omega_M - \frac{1}{\rho}}$. Here, we assume that $\omega_M > \frac{1}{\rho}$. The proof when $\omega_M < \frac{1}{\rho}$ is analogous.

Consider another path $\tau'$ of type $CCC$ with the same initial and the final location as well as the same inflexion points as $\tau$. The only difference is that the laser changes its direction in the middle $C$ segment, i.e., the laser turns anticlockwise in the middle $C$ segment and clockwise in the first and the last $C$ segment in trajectory $\tau'$. Let $\Delta\psi_2'$ denote the the angle that the laser rotates in the middle $C$ segment in trajectory $\tau'$. As the initial state and the inflexion points are same in both $\tau$ and $\tau'$, the time taken by the Dub-L System in the second $C$ segment must be the same. This implies

$$\Delta\psi_2' = \frac{\Delta\psi_2(\omega_M + \frac{1}{\rho})}{\omega_M - \frac{1}{\rho}} \Rightarrow \Delta\psi_2' > \Delta\psi_2.$$

As $\Delta\psi_2 \geq \pi$, it follows that by changing the direction in which the laser rotates in the middle $C$ segment, the same orientation can be achieved in less time; meaning that by switching off the laser over some non-trivial interval of time in the middle $C$ segment and then rotating it anti-clockwise, the same orientation of the laser is achieved as in trajectory $\tau$ at the second inflexion point. This implies that, for trajectory $\tau'$, $t_l^* > t_0$. From Lemma 63, this further implies that there exists a trajectory of type $\{CS, CC\}$ that requires shorter time than trajectory $\tau'$. Finally, as the time taken by the Dub-L System in trajectory $\tau$ and $\tau'$ is the same, it follows that there exists a trajectory of type $\{CS, CC\}$ that requires shorter time than trajectory $\tau$. Thus, $\tau$ is not optimal. Finally, $\Delta\psi_2 > 0$ as $\omega_M \neq \frac{u}{\rho}$. This concludes the proof. □

**Lemma 66.** *Suppose that a path of type $CCC$ for the Dub-L system exists and $\omega_M = \frac{1}{\rho}$. Then, $0 < \beta < \pi$, where $\beta$ denotes the angles subtended by the middle $C$ segment of the $CCC$ path.*

*Proof.* Denote the path of type $CCC$ as $\tau$ and suppose that $\beta \geq \pi$. Since $\omega_M = \frac{u}{\rho}$ and the vehicle and the laser turn in opposite direction in the middle $C$ segment, it follows that the orientation of the laser, with respect to the positive $x$-axis, at the first and the second inflexion point is equal. Mathematically, $\psi(t^1_{inf}) = \psi(t^2_{inf})$, where $\psi(t^1_{inf})$ (resp. $\psi(t^2_{inf})$) denotes the orientation of the laser at the first and the second inflexion point.

Similar to the proof of Lemma 65, consider another path $\tau'$ of type $CCC$ with the same initial and the final location as well as the same inflexion points as $\tau$. The only difference is that the laser changes its direction in the middle $C$ segment. Let $\Delta\psi'_2$ denote the angle that the laser turns in the middle $C$ segment in trajectory $\tau'$. Then, equating the time taken by the vehicle in the second segment and the time taken by the turret to rotate and using that $\omega_M = \frac{1}{\rho}$ yields

$$\rho\beta = \frac{\rho\Delta\psi'_2}{2} \Rightarrow 2\beta = \Delta\psi'_2.$$

Since $\beta \geq \pi$, $\Delta\psi'_2 \geq 2\pi$ meaning that by having the laser switched off over some non-zero interval of time in $\tau'$, the same orientation of the laser can be achieved as in trajectory $\tau$ at final time. This implies that there exists a shorter trajectory than $\tau'$ and $\tau$ of type $\{CS, CC\}$ and thus, $\tau$ is not optimal. This concludes the proof. □

**Lemma 67.** *No path of type $CCC$ taken by the Dub-L System is optimal.*

*Proof.* The proof is in two parts based on whether $\omega_M = \frac{1}{\rho}$ or not. We start with when $\omega_M \neq \frac{1}{\rho}$.

**Case 1 ($\omega_M \neq \frac{1}{\rho}$):** Let $\tau$ denote a $CCC$ type path for the Dub-L System with angles of the three $C$ segments as $\alpha_0, \beta_0$, and $\gamma_0$, respectively, and without loss of generality, suppose that the laser turns clockwise. Then, the time taken by the Dub-L System or equivalently the length of $\tau$ is $L_\tau = \rho(\alpha_0 + \beta_0 + \gamma_0) = \frac{\Delta\psi_1}{\omega_M + \frac{1}{\rho}} + \frac{\Delta\psi_2}{\omega_M - \frac{1}{\rho}} + \frac{\Delta\psi_3}{\omega_M + \frac{1}{\rho}}$, where $\Delta\psi_1$, $\Delta\psi_2$, and $\Delta\psi_3$ denotes the angle that the laser rotates in the first, second, and third $C$ segment of $\tau$. Let $\epsilon_1 > 0$ and $\epsilon_2 > 0$ denote an infinitesimally small positive real number. We can deform this path into a similar one with same endpoints and angles $\alpha = \alpha_0 + \epsilon_1, \beta = \beta_0 - \epsilon_2$, and $\gamma_0$ so that

$$\begin{aligned}
&\cos(\Delta\psi_1') + \cos(\Delta\psi_1' - \Delta\psi_2') = \cos(\Delta\psi_1) + \\
&\cos(\Delta\psi_1 - \Delta\psi_2) \text{ and} \\
&\sin(\Delta\psi_1') + \sin(\Delta\psi_1' - \Delta\psi_2') = \sin(\Delta\psi_1) + \\
&\sin(\Delta\psi_1 - \Delta\psi_2),
\end{aligned} \tag{6.8}$$

where $\Delta\psi_1' = \Delta\psi_1 + \rho\epsilon_1(\omega_M + \frac{1}{\rho})$, $\Delta\psi_2' = \Delta\psi_2 - \rho\epsilon_2(\omega_M - \frac{1}{\rho})$ and are obtained by using the fact that the time taken by the vehicle in the first (resp. second) segment of the deformed path is $\rho(\alpha_0 + \epsilon_1) = \frac{\Delta\psi_1'}{\omega_M + \frac{1}{\rho}}$ (resp. $\rho(\beta_0 - \epsilon_2) = \frac{\Delta\psi_2'}{\omega_M - \frac{1}{\rho}}$). The determinant of the Jacobian matrix obtained from the set of equations in (6.8) with respect to $\epsilon_1$ and $\epsilon_2$ is $\rho^2\omega_M^2 \sin(\Delta\psi_2 - \epsilon_2)$. From Lemma 65, it follows that the determinant of the Jacobian is not zero near $\Delta\psi_2$, and thus, such a deformation is possible.

Now, consider one such deformation with $\epsilon_1 = \epsilon_2 = \epsilon$, where $\epsilon > 0$ is a very small positive real number and let $\tau'$ denote the trajectory obtained though this deformation. Although $\tau'$ may have the laser's orientation different than that of trajectory $\tau$, note that $L_\tau = L_{\tau'}$, where $L_{\tau'}$ denotes the length of trajectory $\tau'$. This implies that both trajectories end at the same time $t_f$. Further, as the laser turns clockwise, we have

$$\psi_2' - \psi_2 = \psi_1' - \psi_1 + \rho\epsilon(\omega_M - \tfrac{1}{\rho})$$
$$\Rightarrow \psi_2' - \psi_2 = -2\epsilon,$$

where $\psi_1'$ and $\psi_2'$ (resp. $\psi_1$ and $\psi_2$) denote the orientation of the laser at the first and the second inflexion point, respectively, in trajectory $\tau'$ (resp. $\tau$). Let $\psi_\tau(t_f)$ and $\psi_{\tau'}(t_f)$ denote the orientation of the laser at final time $t_f$ in trajectory $\tau$ and $\tau'$, respectively. Let $T_1 = \frac{\Delta\psi_1}{\omega_M + \frac{1}{\rho}}$ and $T_2 = \frac{\Delta\psi_2}{\omega_M - \frac{1}{\rho}}$. Since $L_\tau = L_{\tau'}$ and using the fact that the laser turns clockwise, we obtain

$$T_1 + T_2 + \frac{\psi_2 - \psi(t_f)}{\omega_M + \frac{1}{\rho}} = T_1 + T_2 + \frac{\psi_2' - \psi'(t_f)}{\omega_M + \frac{1}{\rho}}$$
$$\Rightarrow \psi'(t_f) - \psi(t_f) = \psi_2' - \psi_2 < 0,$$

which is a contradiction as $\psi_{\tau'}(t_f) \geq \psi_\tau(t_f)$ must hold if trajectory $\tau$ is optimal, given that the laser is turning clockwise. This is because as $\psi'(t_f) - \psi(t_f) < 0$, $\psi'(t_f) = \psi(t_f)$ can be achieved by switching the laser off over some non-trivial interval of time. From Lemma 63, this implies that there exists a $CS$ or a $CC$ type trajectory from the same initial location which requires less time than trajectory $\tau'$ ( and equivalently $\tau$). Thus, $\tau$ is not optimal and the result is established.

**Case 2 ($\omega_M = \frac{1}{\rho}$):** As the proof of this case is analogous to Case 1, we only provide an outline of the proof for brevity. Consider the same trajectories $\tau$ and $\tau'$ as in Case 1 and let $\epsilon_1$ and $\epsilon_2 > 0$ denote an infinitesimally small positive real number so that

$$\begin{aligned} \cos(\alpha) + \cos(\alpha - \beta) &= \cos(\alpha_0) + \cos(\alpha_0 - \beta_0), \\ \sin(\alpha) + \sin(\alpha - \beta) &= \sin(\alpha_0) + \sin(\alpha_0 - \beta_0). \end{aligned} \tag{6.9}$$

Then, the determinant of the Jacobian matrix obtained from the set of equations in (6.9) with respect to $\epsilon_1$ and $\epsilon_2$ is $\sin(\beta_0 - \epsilon_2)$ which, from Lemma 66 is not equal to zero near $\beta_0$. Thus, such a deformation is possible.

Since $\omega_M = \frac{1}{\rho}$ and the vehicle and the laser turns in the opposite direction in the middle $C$ segment in both $\tau$ and $\tau'$, it follows that $\psi_2 = \psi_1$ and $\psi_2' = \psi_1'$ holds, respectively. Using

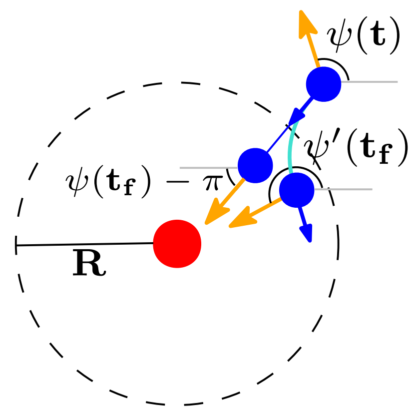



Figure 6.3 Illustration of Case 1 of proof of Lemma 69. The blue line denotes trajectory $\tau$ and the cyan curve denotes trajectory $\tau'$.

$L_\tau = L_{\tau'}$ and that the laser turns clockwise, we obtain

$$T_1 + T_2 + \frac{\psi_2 - \psi(t_f)}{\omega_M + \frac{1}{\rho}} = T_1 + T_2 + \frac{\psi_2' - \psi'(t_f)}{\omega_M + \frac{1}{\rho}}$$
$$\Rightarrow \psi'(t_f) - \psi(t_f) = \psi_1' - \psi_1 < 0.$$

Analogous to Case 1, this is a contradiction and thus, the result is established. □

We now establish the main result of this work.

**Theorem 68.** *The optimal path of the Dub-L system is of type $\{CSC, CC\}$ or a subsegment of it.*

*Proof.* If an optimal trajectory of the Dub-L System contains a straight line segment, it follows from Corollary 11 that the trajectory must be of type $CSC$ or a degenerate form if it. Further, from Lemma 67, since a path of type $CCC$ is not optimal, it follows that the optimal path of the Dub-L System is of type $\{CSC, CC\}$ or a subsegment of it. This concludes the proof. □

Through the next two results, we establish that an optimal path, except a path of type $C$, ends at a distance $R$ from the target.

**Lemma 69.** *For any optimal trajectory of the Dub-L System that consists of a straight line, $x^2(t_f) + y^2(t_f) = R^2$ holds at time $t_f$.*

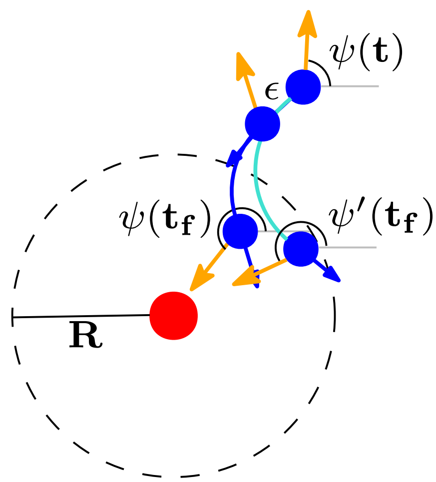



Figure 6.4 Illustration of Case 2 of proof of Lemma 69. The blue line denotes trajectory $\tau$ and the cyan curve denotes trajectory $\tau'$.

*Proof.* Since the trajectory of the Dub-L system may end with an arc or a straight line, we consider two cases based on whether the trajectory ends with an $S$ or a $C$. For both cases, we assume that the laser is turning anti-clockwise. The proof when the laser turns clockwise is analogous.

**Case 1 (Trajectory is of $S$ type at $t_f$):** Consider a trajectory $\tau$, as shown in Figure 6.3, in which the Dubins vehicle moves on a straight line until time instant $t_f$ and for which $x^2(t_f) + y^2(t_f) < R^2$ holds. Consider another trajectory $\tau'$ that starts from the same initial condition as $\tau$ and has the vehicle turn anti-clockwise at time $t = t_f - \epsilon$, where $\epsilon > 0$ is a very small positive real number, such that the length of both $\tau$ and $\tau'$ is equal. As the laser is turning anti-clockwise, from Lemma 61, the vehicle must turn anti-clockwise in $\tau'$ in the final segment. Let $\psi(t)$ be the orientation of the laser at time instant $t$. Since both $\tau$ and $\tau'$ have the same initial condition, it follows that the orientation of the laser is same for both trajectories at time $t$. Let $\psi(t_f)$ (resp. $\psi'(t'_f)$) denote the orientation of the laser at time $t_f$ (resp. $t'_f$) in trajectory $\tau$ (resp. $\tau'$). Analogous to the proof of Lemma 67, as the length, or equivalently the time taken by both trajectories is equal,

$$\frac{\psi'(t'_f) - \psi(t)}{\omega_M + \frac{1}{\rho}} = \frac{\psi(t_f) - \psi(t)}{\omega_M}$$
$$\Rightarrow \psi'(t'_f) - \psi(t_f) = \frac{\psi(t_f) - \psi(t)}{\rho} > 0.$$

This is a contradiction as this implies that $\psi'(t'_f) = \psi(t_f)$ can be achieved by having the laser switched off over some non-trivial interval of time implying that there exists a trajectory that requires length shorter than $\tau'$ which turns the vehicle until some time strictly less than $t_f$.

**Case 2 (Trajectory is of $C$ type at $t_f$):** Consider a trajectory $\tau$ (cf. Fig. 6.4) which ends with a circular arc at time instant $t_f$ and for which $x^2(t_f)+y^2(t_f) < R^2$ holds. Consider another trajectory $\tau'$ that starts from the same initial condition as $\tau$ and that has the vehicle turn anti-clockwise at time $t = t^2_{inf} - \epsilon$, where $t^2_{inf}$ denotes the time the vehicle turns anti-clockwise in trajectory $\tau$, i.e., the time instant when the vehicle reaches the second inflexion point in $\tau$. As the time taken by the vehicle is equal in both trajectories, we obtain

$$\frac{\psi'(t'_f) - \psi(t)}{\omega_M + \frac{1}{\rho}} = \epsilon + \frac{\psi(t_f) - \psi(t) - \epsilon\omega_M}{\omega_M + \frac{1}{\rho}}$$
$$\Rightarrow \psi'(t'_f) - \psi(t_f) = \frac{\epsilon}{\rho} > 0$$

which, analogous to Case 1, implies that there exists a trajectory of length shorter than $\tau'$ and $\tau$. This concludes the proof. □

**Lemma 70.** *If the trajectory of the Dub-L System is of type $CC$, then $x^2(t_f) + y^2(t_f) = R^2$ holds at time $t_f$.*

*Proof.* The proof is omitted for brevity as it is analogous to that of Lemma 69. □

### 6.5 Solution for Optimal Trajectory

In this section, we characterize the trajectories of the Dub-L System. We start by characterizing the trajectories when $t^*_l > t_0$.

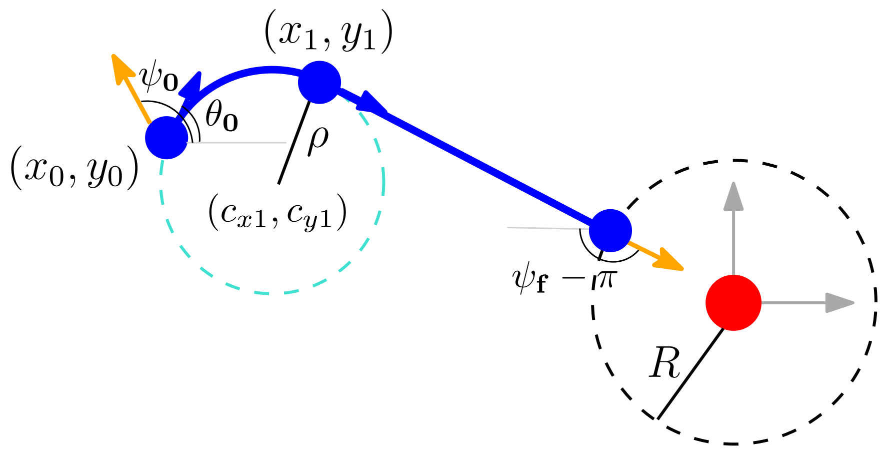



Figure 6.5 An illustration of $RS$ type trajectory when $t_l^* > t_0$ with the laser turning clockwise.

### 6.5.1 Solution for trajectories when $p_\psi = 0$

Recall, from Lemma 63 and Corollary 13, that the optimal trajectory ends at a distance $R$ from the target and is of type $\{CS, CC\}$ or a subsegment of it. Since the location and the orientation of the Dubins vehicle at final time can be determined through elementary geometry, we only describe the solution of $CS$ type trajectory as the solution for the $CC$ type trajectory is analogous.

Given the initial state of the Dub-L System (refer to Fig. 6.5), the center $(c_{x1}, c_{y1})$ of the circle formed by the $C$ segment of the $CS$ trajectory is $(x_0 + \rho\sin(\theta_0), y_0 - \rho\cos(\theta_0))$. Note that Figure 6.5 is only an illustration and the description for an $LS$ trajectory is analogous. Then, from Corollary 11, as the $S$ segment passes through the target location, it follows that the equation of the $S$ segment is $y = \tan(\theta(t_f))x$. As the orientation of the vehicle does not change in the $S$ segment, $\theta(t_f)$ can be determined by determining the orientation of the vehicle at the inflexion point. In particular, $\theta(t_f)$ is determined by finding the distance from the center $(c_{x1}, c_{y1})$ to the $S$ segment at the inflexion point. Mathematically, this yields

$$\frac{|c_{y1} - \tan(\theta(t_f))c_{x1}|}{\sqrt{\tan^2(\theta(t_f)) + 1}} = \rho,$$

$$\Rightarrow \tan(\theta(t_f)) = \frac{c_{y1}c_{x1} \pm \rho\sqrt{c_{y1}^2 + c_{x1}^2 - 1}}{c_{x1}^2 - \rho^2}.$$

Note that, for a specified $\theta_0$, one out of the two values of $\theta_f$ can be eliminated. Finally, by determining the intersection between the $S$ segment and a circle of radius $R$, centered at the target location, we obtain $(x(t_f), y(t_f))$ as $(\frac{\pm R}{\sqrt{1+\tan^2(\theta(t_f))}}, \frac{\pm\tan(\theta(t_f))R}{\sqrt{1+\tan^2(\theta(t_f))}})$. We now determine $t_l^*$.

Let $T_D$ denote the time taken by the Dub-L System to move from $(x_0, y_0)$ with orientation $\theta_0$ to the location $(x(t_f), y(t_f))$ with orientation $\theta(t_f)$. The time required by the laser to rotate from $\psi_0$ to $\psi(t_f) = \theta(t_f)$ is $T_L = \frac{\psi_0 - \psi(t_{inf}^1)}{\omega_M + \frac{1}{\rho}} + \frac{\psi_f - \psi(t_{inf}^1)}{\omega_M}$, where $\psi_{t_{inf}}^1$ denotes the orientation of the laser at the inflexion point. Then, $t_l^* = T_D - T_L$.

We now characterize the solution of trajectories when $p_\psi \neq 0$.

### 6.5.2 Solution for trajectories when $t_l^* = t_0$

In this section, we restrict to the case when $p_0 > 0$ under the conjecture, similar to the work in [44], that the class of trajectories obtained when $p_0 = 0$ is a subset of $\{CSC, CC\}$. Thus, we normalize all other costates by $p_0$. Further, for brevity, we only describe the solution for $CC$ and $CSC$ trajectories and omit the description of the subsegments as the process is analogous.

From Lemma 69 and Lemma 70, as the condition $x^2(t_f)+y^2(t_f) = R^2$ holds for both $CSC$ and $CC$ trajectory, we have that $x(t_f) = R\cos(\eta)$ and $y(t_f) = R\sin(\eta)$, where $\eta$ denotes the angle to the final location of the Dub-L System measured from the positive $x$-axis. Further, from the transversality conditions, we obtain

$$c_x = \lambda_1 R\cos(\eta) - \frac{c_0\sin(\eta)}{R}, c_y = \lambda_1 R\sin(\eta) + \frac{c_0\cos(\eta)}{R}. \tag{6.10}$$

We now establish the solution for the $CC$ type trajectory followed by the $CSC$ type trajectory.

#### 6.5.2.1 $CC$ type trajectory

Let the location and the orientation of the Dub-L System at the inflexion point be $x(t_{inf}^1), y(t_{inf}^1)$, and $\theta(t_{inf}^1)$, respectively. Further, let $(c_{x1}, c_{y1})$ and $(c_{x2}, c_{y2})$ denote the centers of the two circles of the $CC$ segment (Fig. 6.6). Then, as $\theta(t_{inf}^1)$ is perpendicular to the line joining the two centers $(c_{x1}, c_{y1})$ and $(c_{x2}, c_{y2})$, we obtain

$$\tan(\theta_1) = -\frac{c_{x2} - c_{x1}}{c_{y2} - c_{y1}},$$

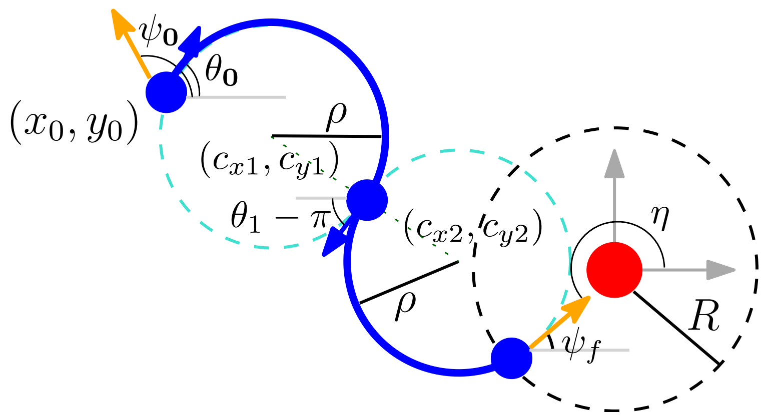



Figure 6.6 An illustration of $RL$ type trajectory when $t_l^* = t_0$.

where $(c_{x1}, c_{y1})$ is $(x_0 + \rho\sin(\theta_0), y_0 - \rho\cos(\theta_0))$, $(c_{x2}, c_{y2})$ is $(x(t_f) - \rho\sin(\theta(t_f)), y(t_f) + \rho\cos(\theta(t_f)))$. Further, $(x(t_{inf}^1), y(t_{inf}^1))$ can be written as $(c_{x2}+\rho\sin(\theta(t_{inf}^1)), c_{y2}-\rho\cos(\theta(t_{inf}^1)))$. Next, as the Hamiltonian must be zero at the initial location, inflexion point, and the final location, we obtain

$$H_0 = 1 + c_x\left(\cos(\theta_0) + \frac{y_0 u_i^*}{\rho}\right) + c_y\left(\sin(\theta_0) - \frac{x_0 u_i^*}{\rho}\right) + c_\psi \omega_M^* = 0. \tag{6.11}$$

$$H_{inf}^1 = 1 + c_x\left(\cos(\theta(t_{inf}^1)) + \frac{y(t_{inf}^1)\sin(\theta(t_{inf}^1))}{x(t_{inf}^1)}\right) + c_\psi \omega_M^* = 0. \tag{6.12}$$

$$H_f = 1 + c_x\cos(\theta(t_f)) + c_y\sin(\theta(t_f)) + c_\psi\left(\frac{u_f^*}{\rho} + \omega_M^*\right) = 0, \tag{6.13}$$

where we have used that $p_\theta(t_f) = 0$ and $p_\theta + p_\psi = 0$ at the inflexion points. Next, from Fig. 6.6 and Lemma 61, as the vehicle turns anti-clockwise in the final $C$ segment, the laser must turn anti-clockwise during the entire trajectory. Finally, let $T_D = \rho(\alpha+\beta)$ be the time taken by the Dub-L system to move from the initial location and orientation to the final location and orientation, where $\alpha$ and $\beta$ denotes the angle subtended by the $C$ segments. Further, let $T_L = \frac{\psi_0 - \psi(t_{inf}^1)}{\omega_M - \frac{1}{\rho}} + \frac{\psi(t_{inf}^1) - \psi(t_f)}{\omega_M + \frac{1}{\rho}}$ be the time taken by the laser to orient to the final orientation $\psi(t_f) = \eta + \pi$. Then, as the time taken by the laser and the Dubins vehicle must be the

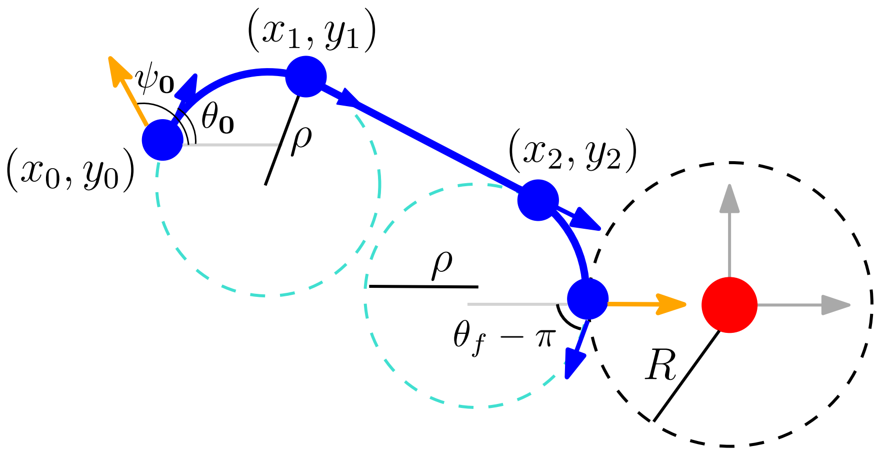



Figure 6.7 An illustration of $RSR$ type trajectory when $t_l^* = t_0$.

same, we obtain

$$T_D - T_L = 0. \tag{6.14}$$

Note that the angles $\alpha$ and $\beta$ can be determined and expressed in terms of $\eta$ and $\theta(t_f)$ by using Law of Cosines. Thus, by substituting $c_x$ and $c_y$ from equation (6.10) in equations (6.11)-(6.14) yields a set of four equations with four unknowns $\eta, \lambda_1, \theta_f, c_0$. Solving these equations yield $\theta_f, \psi_f, x_f, y_f$.

#### 6.5.2.2 *CSC* type trajectory

We now characterize the solution of the $CSC$ trajectory of the Dub-L System.

Analogous to the $CS$ trajectory is subsection 6.5.1, the first inflexion point and the orientation of the Dub-L System at the first inflexion point can be determined through elementary geometry. Then, as the Hamiltonian must be 0 for all time $t \in [t_0, t_f]$, equating the Hamiltonian at the initial $H_0$, first inflexion point $H^1_{inf}$, second inflexion point $H^2_{inf}$, and the final location $H_f$ yields the following set of equations.

$$\begin{aligned} H_0 = 1 + c_x \left(\cos(\theta_0) + \frac{y_0 u_i^*}{\rho}\right) + c_y \left(\sin(\theta_0) - \frac{x_0 u_i^*}{\rho}\right) + \\ c_\psi \omega_M^* = 0 \end{aligned} \tag{6.15}$$

$$H_{inf}^1 = 1 + c_x \left( \cos(\theta(t_{inf}^1)) + \frac{y(t_{inf}^1)\sin(\theta(t_{inf}^1))}{x(t_{inf}^1)} \right) + c_\psi \omega_M^* = 0. \tag{6.16}$$

$$H_{inf}^2 = 1 + c_x \left( \cos(\theta(t_{inf}^1)) + \frac{y(t_{inf}^2)\sin(\theta(t_{inf}^1))}{x(t_{inf}^2)} \right) + c_\psi \omega_M^* = 0. \tag{6.17}$$

$$H_f = 1 + c_x \cos(\theta(t_f)) + c_y \sin(\theta(t_f)) + c_\psi(\frac{u_f^*}{\rho} + \omega_M^*) = 0 \tag{6.18}$$

where we have $p_\theta(t_f) = 0$ and $p_\theta + p_\psi = 0$ at the inflexion points and $\theta_1 = \theta_2$ (Fig. 6.7). Further, location $(x(t_{inf}^2), y(t_{inf}^2))$ can be expressed in terms of $(x(t_f), y(t_f))$ as $(R\cos(\eta) - \rho\sin(\theta(t_f)) + \rho\sin(\theta(t_{inf}^1)), R\sin(\eta) + \rho\cos(\theta(t_f)) - \rho\cos(\theta(t_{inf}^1)))$. Finally, substituting $c_x$ and $c_y$ from equation (6.10) in equations (6.15)-(6.18) yields a system of four equations with four unknowns, solving which yields the final location for the vehicle. If multiple solutions of $x(t_f), y(t_f)$, and $\theta(t_f)$ are obtained by solving equations (6.15)-(6.18) then, by using the terminal constraint that the laser must be oriented towards the target (equation (6.1)), the optimal location at time $t_f$ can be determined.

### 6.6 Numerical Simulations

In this section, we present some numerical simulations to illustrate the properties characterized in this work. For all of our simulations, $R = \rho = 1$.

Figure 6.8 depicts the shortest path of a $CS$ trajectory when $t_l^* > t_0$, where the final location and the orientation of the Dub-L System is determined according to 6.5.1. The initial state, i.e., $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ are set to $2, 2, \frac{\pi}{2}, \pi$, and 0.3 respectively. It is evident that the condition $x^2(t_f) + y^2(t_f) = R^2$ holds and the inflexion point and the straight line segment are collinear with the target location.

Figure 6.9 shows a $C$ type trajectory with $t_l^* = t_0$ with $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ as $0.5, 0.5, \frac{\pi}{2}, \pi$, and 0.01 respectively. Recall that a $C$ type trajectory is the only trajectory for which the condition $x^2(t_f) + y^2(t_f) = R^2$ may not hold when $t_l^* = t_0$. To determine the final location

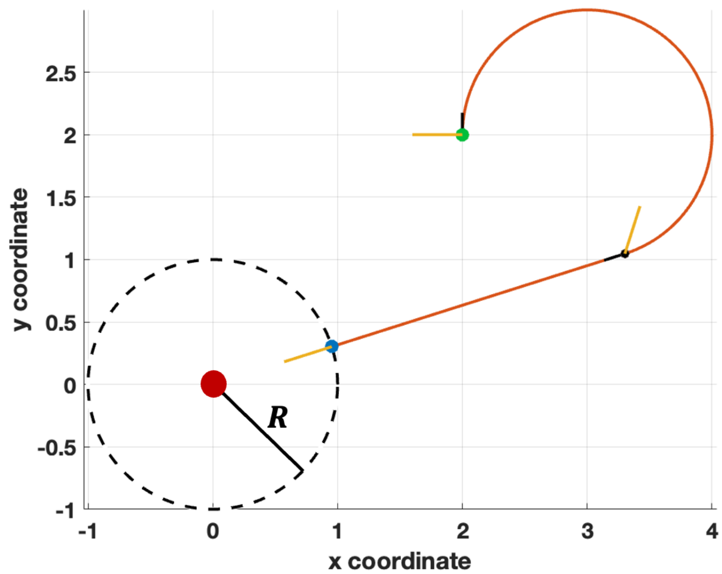



Figure 6.8 A $CS$ type trajectory when $t_l^* > t_0$ with $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ as $2, 2, \frac{\pi}{2}, \pi$, and 0.3 respectively. The black dashed circle denotes the circle of radius $R$ around the target. The initial location is denoted as a green dot and the final location is denoted as the blue dot. The yellow line denotes the laser.

and the orientation of the Dub-L System, we use *fsolve* function in MATLAB to obtain the solution of equations (6.11),(6.13), and (6.14) characterized in Subsection 6.5.2.

Finally, Figure 6.10 shows a $RSR$ trajectory when $t_l^* = t_0$ with $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ as $2, 2, \frac{\pi}{2}, \frac{4\pi}{3}$, and 0.01 respectively. It is evident that the condition $x^2(t_f) + y^2(t_f) = R^2$ holds and the inflexion point and the straight line segment are collinear with the target location.

## 6.7 Conclusions

We considered a joint motion planning problem of a Dubins-Laser System which consists of a Dubins vehicle and a laser that is attached to the vehicle. The laser has a finite range and can rotate either clockwise or anti-clockwise with a bounded angular speed. The environment consists of a single static target. We characterized various geometric properties of the shortest path of the Dubins-Laser System, including the cooperation between the laser and the Dubins vehicle. We established that the shortest path for the Dubins-Laser System is of type $\{CSC, CC\}$ or a subsegment of it and characterized the solution of each of the 13 candidate trajectories for the shortest path.

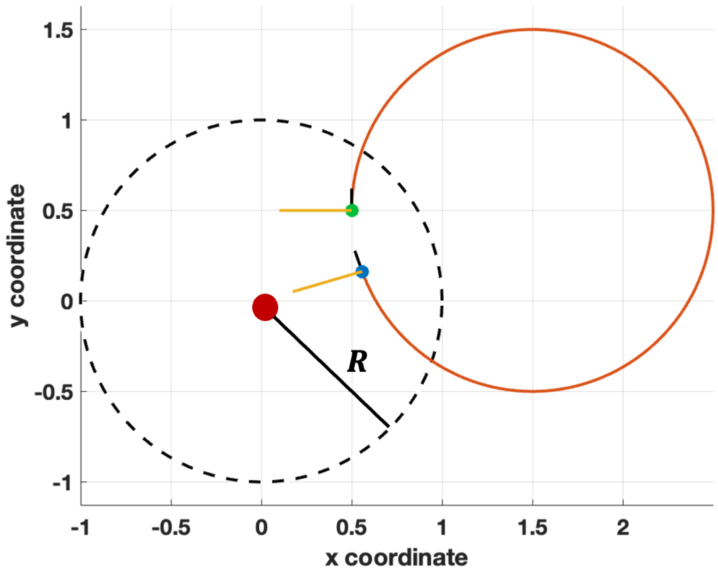



Figure 6.9 A $C$ type trajectory when $t_l^* = t_0$ with $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ as $0.5, 0.5, \frac{\pi}{2}, \pi$, and 0.01 respectively. The black dashed circle denotes the circle of radius $R$ around the target. The initial location is denoted as a green dot and the final location is denoted as the blue dot. The yellow line denotes the laser.

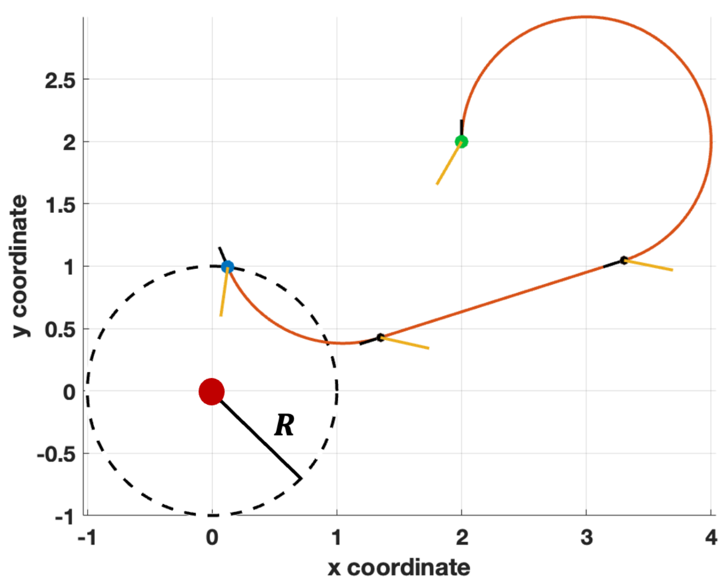



Figure 6.10 A $CSC$ type trajectory with $x_0, y_0, \theta_0, \psi_0$, and $\omega_M$ as $2, 2, \frac{\pi}{2}, \frac{4\pi}{3}$, and 0.01 respectively. The black dashed circle denotes the circle of radius $R$ around the target. The initial location is denoted as a green dot and the final location is denoted as the blue dot. The yellow line denotes the laser.

# CHAPTER 7

# OPTIMAL PURSUIT OF SURVEILLING AGENTS NEAR A HIGH VALUE TARGET

In most of the works on perimeter defense, it is assumed that the location of the perimeter is known to the adversary, which may not be true in all applications. For instance, the location of a high value defense/research facility (target/perimeter) is not precisely known to the adversary. In such scenarios, prior to deploying intruders to breach the perimeter, the adversary will typically obtain the estimates of the location of the facility by deploying adversarial UAVs (or trackers) equipped with some low cost range sensor [14]. To counter these UAVs, the defense system can release mobile pursuers (or defenders) that have the ability to intercept and disable the UAVs or to corrupt the information gathered by obstructing the line-of-sight (cf. Fig. 7.1). These scenarios raise an important question which has not yet been fully explored in the literature – *how does the motion strategy of adversarial trackers change in the presence of one or many pursuers?*

This work is a first step towards formulating an adversarial information gathering problem in presence of a mobile pursuer. Specifically, we introduce a *tracking pursuit* game in a planar environment comprising one single static high value target, a single mobile pursuer and two mobile trackers (adversarial UAVs). We assume that the trackers are slower than the pursuer and can only measure their individual distances from the target. The trackers jointly seek to maximize the tracking performance while simultaneously evading the pursuer at every time instant. On the other hand, the pursuer seeks to capture one of the trackers to hinder the tracking objective of the trackers. Although we considers a planar environment and range-only measurements, we also show how this work can be extended to other sensing models such as bearing measurements.

### 7.0.1 Related Works

General target tracking problems involve a static/moving target whose state (e.g., the position and velocity) needs to be estimated by trackers using measurements based on the

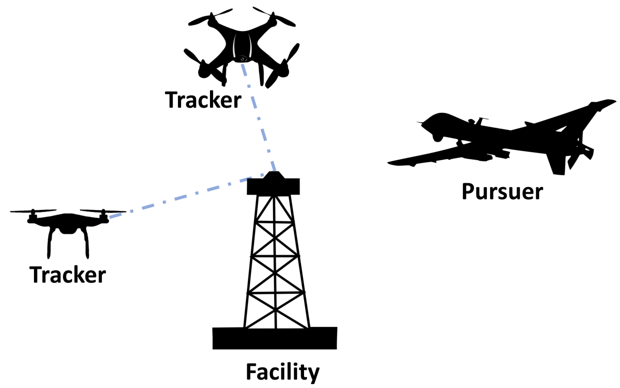



Figure 7.1 Problem Description. Adversarial UAVs (trackers) move to maximize the information gathered using the distance measurements to the facility (target) while simultaneously evading from the pursuer.

distance or bearing or both [25, 13, 55, 60]. A generic approach in these works is to optimize a measure (e.g., trace or determinant) of the estimation error covariance matrix obtained from an Extended Kalman Filter (EKF) used to estimate the state. We refer to [23] and the references therein for the application of state estimation to various target tracking scenarios. Since the relative geometry of the target and the trackers plays an important role in the tracking process, many works [52, 13, 85, 86, 72] have focused on identifying such geometries and motion strategies that optimize the tracking performance such as the determinant of the Fisher Information Matrix (FIM) or the trace of the estimation error covariance of the EKF. Tracking based on metrics of observability have also been considered [26, 65, 66].

All of the above mentioned works only focus on determining optimal trajectories for the trackers to optimize a certain tracking performance, but do not consider the presence of a pursuer. Authors in [40] design strategies for the pursuers to optimize the tracking performance while maintaining a desired formation. In [73], an adaptive sampling approach is considered to track mobile targets and maintain them in the field of view. Authors in [2] propose an algorithm based on rapidly exploring random trees for pursuers to detect and track a target.

Pursuit of mobile agents (or evaders) in the presence of a target has been extensively studied as a differential game known as Target-Attacker-Defender (TAD) games [84, 33, 28, 32]. In these works, the attacker tries to capture a target while simultaneously evading a defender. The objective in these works is to determine optimal cooperative strategies for the the target and the defender to delay the time taken to capture or evade the attacker. This work differs from the aforementioned TAD games as the trackers do not seek to capture the target. Instead, through the measurements obtained, the trackers aim to maximize the information gathered about the target while, evading the pursuer. Another variant of pursuit evasion games is pursuit tracking [74, 57, 87] where the objective of the pursuer is to track the evader by maintaining a fixed distance or Line of Sight to it. In contrast, in this work, the pursuer seeks to capture the trackers which are tracking a static target. The content of this chapter is based on [4].

### 7.0.2 Preliminaries and Contributions

Recall that one of the objectives of the trackers is to maximize the information obtained from the set of range measurements to the target. This motivates the use of Fisher Information Matrix (FIM). The FIM is a symmetric, positive definite matrix that characterizes the amount of information provided by the measurements for the position of the target that is to be estimated. In other words, by moving to locations that provide the highest information, the trackers aim to improve the outcome of the estimation process. Maximizing the FIM can be achieved by maximizing a real-valued scalar function (or a metric) of the FIM. The most commonly used metrics are the trace, determinant and the eigenvalues of the FIM, also known as the A-optimality, D-optimality and E-optimality criteria, respectively [83]. Although the trace of the FIM is easy to compute, we consider the determinant as a metric to be maximized by the trackers. This is because the trace of FIM may be non zero even when the FIM is singular, implying that optimizing the trace of the FIM can result in singular configurations.

In this chapter, we seek to understand the role of a pursuer in tracking problems. Equivalently, we aim to understand how the cost of evasion combined with the tracking cost affects the trajectories, and consequently, the payoff of the trackers. In particular, we consider an instantaneous two player zero sum game between the pursuer and the trackers wherein the pursuer seeks to minimize the (square of) distance to one of the trackers at every time instant whereas the trackers aim to jointly maximize a weighted combination of the determinant of the FIM at every time instant and the distance from the pursuer. Our main contributions are as follows:

1. **Tracking-Pursuit Game with a target:** We introduce a tracking-pursuit problem, modelled as a zero sum game, in a planar environment which consists of a single mobile pursuer, two mobile trackers and a single static target. For ease of presentation, we assume that the tracking agents can only measure the distance to the target and are assumed to be slower than the pursuer. At every time instant, the pursuer aims to minimize the square of the distance to a tracker, whereas the trackers aim to jointly maximize a weighted combination of the determinant of the FIM and the square of the distance to the pursuer from the nearest tracker. The game terminates when the pursuer captures a tracker.

2. **Computing Nash Equilibrium Strategies:** We first establish the optimal strategy for the pursuer. Although the payoff for the trackers is a non-convex function, we show that the optimization problem can be converted to a Quadratically Constrained Quadratic Program (QCQP). We further establish that the optimal strategies obtained for the pursuer and the trackers form a Nash equilibrium of this game.

3. **Numerical Insights:** We provide several numerical examples highlighting the trajectories of the mobile agents and the affect on the instantaneous payoff. In particular, we show that due to the presence of pursuer the determinant of the FIM achieves a lower value. We also show, through one of the examples, that the pursuer can capture a tracker even when the tracker is faster than the pursuer.

4. **Extension to multiple trackers and targets:** Finally, we thoroughly describe how this work extends to the scenarios when there are multiple trackers or multiple targets.

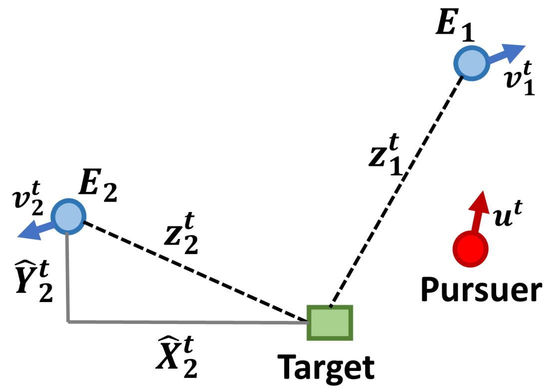



Figure 7.2 Problem Description with $\alpha = 1$. The trackers and the pursuer is denoted by the blue and the red circles, respectively. The (static) high value target is denoted by a green square.

This chapter is organized as follows. Section 7.1 comprises the formal problem definition. In section 7.2, we derive optimal strategies for the pursuer and the trackers, Section 7.3 provides several numerical insights into the problem and Section 7.4 describes the extension of this work to the case of multiple targets and trackers. Finally, Section 7.5 summarizes this work and outlines future directions for this work.

## 7.1 Problem Description

We consider a *tracking evasion* problem in a planar environment which consists of a single static target, a single mobile pursuer and two mobile trackers. We denote the two trackers as $E_1$ and $E_2$, respectively (cf. Fig. 7.2). Each mobile agent is modelled as a single order integrator with bounded maximum speed. The pursuer is assumed to be faster than the trackers and can move with a maximum speed normalized to unity. At every time instant $t$, each tracker $i$, where $i \in \{1, 2\}$, has access to the range measurements, $z_i^t$, to the target located at $s \triangleq [s_x \quad s_y]' \in \mathbb{R}^2$, where $r'$ denotes the transpose of some vector $r$. Let $e_i^t \triangleq \begin{bmatrix} e_{x,i}^t & e_{y,i}^t \end{bmatrix}' \in \mathbb{R}^2$ (resp., $p^t \triangleq \begin{bmatrix} p_x^t & p_y^t \end{bmatrix}' \in \mathbb{R}^2$) denote the position of the $i^{th}$ tracker (resp. the pursuer) and let $v_i^t$ (resp. $u^t$) denote the $i^{th}$ tracker's (pursuer's) control. Then, the motion model and the measurements are given by

$$
\begin{aligned}
e_i^{t+1} &= e_i^t + v_i^t + w_i^t, \qquad \|v_i^t\| \le \mu_i < 1,\ \forall i \in \{1,2\},\\
p^{t+1} &= p^t + u^t + w_p^t, \qquad \|u^t\| \le 1,\\
z_i^t &= \|s - e_i^t\| + V_i^t,\ \forall i \in \{1,2\},
\end{aligned} \tag{7.1}
$$

where, $V_i^t \sim \mathcal{N}(0, \sigma_V^2), \forall i \in \{1,2\}$ denotes the measurement noise, assumed to be mutually independent, $w_i^t \sim \mathcal{N}(0, \sigma_{w_e})$ as well as $w_p^t \sim \mathcal{N}(0, \sigma_{w_p})$ denotes the process noise.

A tracker is said to be *captured* by the pursuer at time instant $t$, if its location is within a unit distance from the pursuer at time $t$. Note that since the pursuer is faster than both trackers, the pursuer can always capture both trackers successively. However, we will see that the game ends when the pursuer captures any one of the trackers. A strategy for the pursuer is defined as $u^t : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$. Similarly, a strategy for an $i^{th}$ tracker is defined as $v_i^t : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2, \forall i \in \{1,2\}$.

We now determine the expression for the determinant of the FIM. Let $h(s, e_1^t, e_2^t) \triangleq [\|s - e_1^t\| \ \|s - e_2^t\|]'$ denote the measurement vector at time instant $t$ and $\nabla_s \triangleq [\frac{\partial}{\partial s_x} \ \frac{\partial}{\partial s_y}]'$. Then, for a given model (7.1) and a measurement vector $h(s, e_1^t, e_2^t)$, the FIM at time instant $t+1$ is [52]

$$
\begin{aligned}
f(s, e_1^t, e_2^t, v_1^t, v_2^t) &= \frac{1}{\sigma_V^2} (\nabla_s h(s, e_1^t, e_2^t))' (\nabla_s h(s, e_1^t, e_2^t)),\\
&= \frac{1}{\sigma_V^2} \sum_{i=1}^{2} \frac{1}{S_i^2} \begin{bmatrix} (X_i^t - v_{x,i}^t)^2 & (X_i^t - v_{x,i}^t)(Y_i^t - v_{y,i}^t) \\ (X_i^t - v_{x,i}^t)(Y_i^t - v_{y,i}^t) & (Y_i^t - v_{y,i}^t)^2 \end{bmatrix},
\end{aligned}
$$

where $X_i^t = s_x - e_{x,i}^t$ (resp. $Y_i^t = s_y - e_{y,i}^t$) denotes the difference in the $x$ (resp. $y$)-coordinate of the target and the $i^{th}$ tracker (cf. Fig. 7.2) and $S_i = \sqrt{(X_i^t - v_{x,i}^t)^2 + (Y_i^t - v_{y,i}^t)^2}$. Since the trackers do not know the location of the target, we replace $s$ by its estimate $\hat{s}$ which is obtained from a centralized EKF. Thus, we obtain the determinant of the FIM as

$$\begin{aligned}\det(f(\hat{s}, e_1^t, e_2^t, v_1^t, v_2^t)) &= \frac{1}{\sigma_V^2}\Big[\sum_{i=1}^{2}\frac{1}{\hat{S}_i^2}(\hat{X}_i^t - v_{x,i}^t)^2\sum_{i=1}^{2}\frac{1}{\hat{S}_i^2}(\hat{Y}_i^t - v_{y,i}^t)^2 - \\ &\qquad \Big(\sum_{i=1}^{2}\frac{1}{\hat{S}_i^2}(\hat{X}_i^t - v_{x,i}^t)(\hat{Y}_i^t - v_{y,i}^t)\Big)^2\Big] \\ &= \frac{1}{\sigma_V^2\hat{S}_1^2\hat{S}_2^2}((\hat{X}_1^t - v_{x,1}^t)(\hat{Y}_2^t - v_{y,2}^t) - (\hat{Y}_1^t - v_{y,1}^t)(\hat{X}_2^t - v_{x,2}^t))^2, \end{aligned} \tag{7.2}$$

where $\hat{X}_i^t = \hat{s}_x - e_{x,i}^t$ and $\hat{Y}_i^t = \hat{s}_y - e_{y,i}^t$ for all $i \in \{1, 2\}$. Note that the determinant of the FIM for a single tracker is equal to zero implying that the configuration is always singular in the case of one tracker.

At the first time instant ($t = 1$), the pursuer selects the tracker which is closest to the pursuer. This selection is characterized by $\alpha \in \{0, 1\}$. Specifically, if tracker $E_1$ is closest to the pursuer, then $\alpha = 1$. Otherwise, $\alpha = 0$. The pursuer, then selects its control, $u^t$, such that the square of the distance to the selected tracker is minimized at every time instant $t \geq 1$. On the other hand, the trackers jointly select their control at every time instant $t \geq 1$ to maximize a weighted combination of the determinant of the FIM and the square of the distance between the selected tracker and the pursuer. We assume that the trackers have information of the location of the pursuer and thus, the choice of $\alpha$ is known to the trackers. Since the two trackers jointly maximize the payoff, we model the interaction between the trackers and the pursuer as a two player zero sum game with the payoff, at time instant $t + 1$, defined as

$$\begin{aligned} J(v_1^t, v_2^t, u^t) = \det(f(\hat{s}, e_1^t, e_2^t, v_1^t, v_2^t)) + \delta(\alpha\|e_1^t + v_1^t - p^t - u^t\|^2 + \\ (1-\alpha)\|e_2^t + v_2^t - p^t - u^t\|^2), \end{aligned} \tag{7.3}$$

where $\delta \in \mathbb{R}$ is a fixed weight associated with the evasion cost (distance between the pursuer and the selected tracker) and is assumed to be known by all agents. The game terminates when the pursuer captures the selected tracker since the determinant of the FIM is always zero for one tracker. We use $t_f$ to denote the time instant when the game terminates.

We now provide two definitions that will be helproofful in establishing our main result in Section 7.2.

**Definition 71** (Best Response)**.** *For a two player zero sum game with the payoff defined as* $J(\gamma, \sigma)$*, the strategy* $\gamma^* \in \Gamma_1$ *for player 1 (minimizer) is called the best response to player 2's (maximizer) strategy* $\sigma \in \Gamma_2$ *if the following holds*

$$J(\gamma^*, \sigma^*) \leq J(\gamma, \sigma^*), \ \forall \gamma \in \Gamma_1.$$

Note that the best response for the maximizer can be analogously defined.

**Definition 72** (Nash Equilibrium)**.** *Given a strategy* $\gamma \in \Gamma_1$ *for player 1 and a strategy* $\sigma \in \Gamma_2$ *for player 2 in a two player zero sum game with the payoff* $J(\gamma, \sigma)$*, the pair of strategies* $(\gamma^*, \sigma^*)$ *is said to be a saddle-point equilibrium strategy if the following holds.*

$$J(\gamma^*, \sigma) \leq J(\gamma^*, \sigma^*) \leq J(\gamma, \sigma^*), \ \forall \gamma \in \Gamma_1, \ \sigma \in \Gamma_2. \tag{7.4}$$

Observe that equation (7.4) in Definition 72 can be rewritten as [39]

$$J(\gamma^*, \sigma^*) = \min_{\gamma \in \Gamma_1} J(\gamma, \sigma^*), \ \text{and} \ J(\gamma^*, \sigma^*) = \max_{\sigma \in \Gamma_2} J(\gamma^*, \sigma),$$

implying that the pair of strategies $(\gamma^*, \sigma^*)$ form a Nash equilibrium if $\gamma^*$ (resp. $\sigma^*$) is the best response to $\sigma^*$ (resp. $\gamma^*$).

We now formally state our objective for the above model.

**Problem Statement:** The aim of this work is to determine saddle-point strategies $u^{t^*} \in \mathbb{R}^2$ and $\{v_1^{t^*}, v_2^{t^*}\} \in \mathbb{R}^2 \times \mathbb{R}^2$ at every time instant $t < t_f$ such that

$$\max_{v_1^t, v_2^t} \min_{u^t} \ J(v_1^t, v_2^t, u^t) = \min_{u^t} \max_{v_1^t, v_2^t} \ J(v_1^t, v_2^t, u^t)$$

holds subject to the individual agents maximum speed constraints, i.e.,

$$\|u^t\| \leq 1, \ \|v_1^t\| \leq \mu_1 < 1, \ \|v_2^t\| \leq \mu_2 < 1.$$

For the problem to be well-posed, we make the following assumption.

**Assumption 1 (A1)[EKF Convergence]:** There exists a time instant $t_e < t_f$ at which the estimates obtained by the trackers are equal to the true location of the target, i.e., $||\hat{s} - s|| = 0, \forall t \geq t_e$.

### 7.2 Optimal Strategies

In this section, we determine the optimal strategies for the pursuer and the trackers. We start with the optimal strategy of the pursuer followed by that of the trackers.

#### 7.2.1 Optimal Strategy of the Pursuer

Without loss of generality, let $\alpha = 1$ at the first time instant and suppose that $v_1^t$ was known to the pursuer. Then, the pursuer solves the following optimization problem.

$$\min_{u^t} \ \delta(\|e_1^t + v_1^t - p^t - u^t\|)^2, \tag{7.5}$$
$$\text{subject to } \|u^t\| \leq 1$$

where we used the fact that the term $\det(f(\hat{s}, e_1^t, e_2^t, v_1^t, v_2^t))$ is not a function of the pursuer's control $u^t$. It follows directly that the solution to the optimization problem (7.5) is

$$u^{t^*}(v_1^t) = \frac{e_1^t + v_1^t - p^t}{\|e_1^t + v_1^t - p^t\|},$$

meaning that the pursuer moves directly towards the first tracker, $E_1$, with unit speed as long as the tracker's position at time instant $t+1$, i.e., $e_1^t + v_1^t$, is not within a unit distance from the current position of the pursuer. Otherwise, the optimal pursuer strategy is $e_1^t + v_1^t - p^t$, implying that the tracker is guaranteed to be captured (evasion cost is zero) at time instant $t+1 = t_f$. This further implies that the trackers will move only to maximize the determinant of FIM at time instant $t = t_f - 1$ as tracker $E_1$ is guaranteed to be captured at time $t+1$.

Thus, the optimal strategy for the pursuer at every time instant $t < t_f - 1$ is

$$u^{t^*}(v_1^t, v_2^t) = \begin{cases} \frac{e_1^t+v_1^t-p^t}{\|e_1^t+v_1^t-p^t\|}, \text{if } \alpha = 1, \\ \frac{e_2^t+v_2^t-p^t}{\|e_2^t+v_2^t-p^t\|}, \text{ otherwise.} \end{cases} \tag{7.6}$$

Further, the optimal strategy for the pursuer at time instant $t = t_f - 1$ is

$$u^{t^*}(v_1^t, v_2^t) = \begin{cases} e_1^t + v_1^t - p^t, \text{if } \alpha = 1, \\ e_2^t + v_2^t - p^t, \text{ otherwise} \end{cases} \tag{7.7}$$

if $\|e_1^t + v_1^t - p^t\| < 1$ (resp. $\|e_2^t + v_2^t - p^t\| < 1$) for $\alpha = 1$ (resp. $\alpha = 0$). Otherwise, the optimal strategy for the pursuer at time instant $t_f - 1$ is given by equation (7.6). Note that although we assumed that $v_i^t, \forall i \in \{1, 2\}$ was known to the pursuer, in reality, the pursuer does not have this information. This means that the optimal strategy defined in (7.6) is an *anticipatory* strategy of the pursuer based on the belief of the trackers' strategy. As will be clear from the next subsection, we use the optimal strategy, $u^{t^*}(v_1^t, v_2^t)$, of the pursuer to determine the optimal state-feedback strategies of the trackers $v_1^{t^*}$ and $v_2^{t^*}$. Substituting $v_1^{t^*}$ and $v_2^{t^*}$ into (7.6) implies that $u^{t^*}$ is a state-feedback strategy.

In the next section, we determine the optimal strategies of the trackers. Since maximizing only the determinant of the FIM has been extensively studied [52], we only focus on the case that the position of the tracker being pursued at time instant $t_f - 1$ is more than a unit distance from the current position of the pursuer. In other words, the pursuer's optimal strategy is given by equation (7.6) at time $t_f - 1$. Note that the optimal strategy of the pursuer for any time instant $t < t_f - 1$ is given by equation (7.6). Further, for ease of presentation, we drop the dependency on time from the notations in the next subsection.

### 7.2.2 Optimal Strategies of the Trackers

For a given value of $\alpha$, the trackers jointly solve the following optimization problem.

$$\max_{v_1, v_2} \det(f(\hat{s}, e_1, e_2, v_1, v_2)) + \delta(\|e_1 + v_1 - p - u\|)^2$$

$$\text{subject to } \|v_1\| \leq \mu_1, \|v_2\| \leq \mu_2,$$

where, without loss of generality, we assumed that $\alpha = 1$. Substituting $u^*(v_1, v_2)$ from equation 7.6 as well as the expression of the determinant yields

$$\max_{v_1,v_2} \frac{1}{\sigma_V^2 \hat{S}_1^2 \hat{S}_2^2}\Big((\hat{X}_1 - v_{x,1})(\hat{Y}_2 - v_{y,2}) - (\hat{Y}_1 - v_{y,1})(\hat{X}_2 - v_{x,2})\Big)^2 + \delta(\|e_1 + v_1 - p\| - 1)^2, \tag{7.8}$$

$$\text{subject to } \|v_1\| \le \mu_1, \ \|v_2\| \le \mu_2.$$

Although the constraints are convex, the objective is a non-convex function of $v_i, \forall i \in \{1, 2\}$ and thus, computing a global maximizer is difficult. In what follows, we show that this optimization problem is equivalent to solving a quadratically constrained quadratic program (QCQP) [17].

For ease of presentation, we use the following notation in the next result.

Let $V = \begin{bmatrix} v_{x,1} & v_{y,1} & v_{x,2} & v_{y,2} & v_{x,1}v_{y,2} & v_{x,2}v_{y,1} \end{bmatrix}' \in \mathbb{R}^6$ and let

$$z^2 = \frac{1}{\hat{S}_1^2 \hat{S}_2^2}\big((\hat{X}_1 - v_{x,1})(\hat{Y}_2 - v_{y,2}) - (\hat{Y}_1 - v_{y,1})(\hat{X}_2 - v_{x,2})\big)^2.$$

**Lemma 73.** *Suppose $\alpha = 1$ and let $m = \|e_1 + v_1 - p\|$. Then, the optimization problem defined in* (7.8) *is equivalent to solving a QCQP given by*

$$\begin{aligned} &\max_{\tilde{V}} \tilde{V}' P \tilde{V} \\ &\textit{subject to} \\ &\tilde{V}' Q_j \tilde{V} \le 0, \forall j \in \{1, 2\} \\ &\tilde{V}' F \tilde{V} = 0, \\ &\tilde{V}' M_1 \tilde{V} = 0, \\ &\tilde{V}' L_g \tilde{V} = 0, \forall g \in \{1, \dots, 10\} \\ &\tilde{V}_8 = 1, \end{aligned} \tag{7.9}$$

*where* $\tilde{V} \triangleq \begin{bmatrix} V' & m & 1 & zV' & zv_{x,1}v_{x,2} & zv_{y,1}v_{y,2} & z \end{bmatrix}' \in \mathbb{R}^{17}$, *$\tilde{V}_k$ denotes the $k^{th}$ entry of vector $\tilde{V}$ and the matrices $P, M_1, Q_j, \forall j \in \{1, 2\}$ and $L_g, \forall g \in \{1, \dots, 10\}$ are as defined in the Appendix.*

*Proof.* By replacing $\|e_1 + v_1 - p\|$ by $m$, the optimization problem defined in (7.8) can be rewritten as

$$\max_{v_1,v_2,m,z} \ \frac{1}{\sigma_V^2} z^2 + \delta(m-1)^2$$
$$\text{subject to } \|v_1\| \leq \mu_1,\ \|v_2\| \leq \mu_2,\ m^2 = \|e_1 + v_1 - p\|^2,$$
$$\Big((\hat{X}_1 - v_{x,1})(\hat{Y}_2 - v_{y,2}) - (\hat{Y}_1 - v_{y,1})(\hat{X}_2 - v_{x,2})\Big)^2 - z^2 \hat{S}_1^2 \hat{S}_2^2 = 0.$$

Observe that the optimization problem is now a polynomial in the original optimization variables and the additional variables $z$ and $m$. By adding some extra variables corresponding to the terms that are polynomial in the optimization variables $v_{x,i}$ and $v_{y,i}, \forall i \in \{1, 2\}$, we now convert the aforementioned optimization problem into a QCQP.

Let $V = \begin{bmatrix} v_{x,1} & v_{y,1} & v_{x,2} & v_{y,2} & v_{x,1}v_{y,2} & v_{x,2}v_{y,1} \end{bmatrix}' \in \mathbb{R}^6$. Then, we define a vector of optimization variables $\tilde{V} \in \mathbb{R}^{17}$ as

$$\tilde{V} = \begin{bmatrix} V' & m & 1 & zV' & zv_{x,1}v_{x,2} & zv_{y,1}v_{y,2} & z \end{bmatrix}'.$$

Taking the square on both sides of the norm constraints, the above optimization problem yields the QCQP form as defined in (7.9). Note that the constraint $\tilde{V}'M_1\tilde{V} \equiv \|e_1 + v_1 - p\|^2 - m^2$. Further, the set of constraints $\tilde{V}'L_g\tilde{V} = 0, \forall g \in \{1, \ldots, 10\}$ characterize the relationship between the elements of $\tilde{V}$. As described in [50], the equality constraints in optimization problem (7.9) can be replaced with two inequality constraints ,thus, reducing the optimization problem in the standard QCQP form. This concludes the proof. □

Following similar steps, an analogous optimization problem when $\alpha = 0$ is

$$\max_{\tilde{V}} \tilde{V}' P \tilde{V}$$

subject to

$$\begin{aligned}
&\tilde{V}' Q_j \tilde{V} \leq 0, \forall j \in \{1, 2\} \\
&\tilde{V}' M_0 \tilde{V} = 0, \\
&\tilde{V}' F \tilde{V} = 0, \\
&\tilde{V}' L_g \tilde{V} = 0, \forall g \in \{1, \ldots, 10\} \\
&\tilde{V}_8 = 1,
\end{aligned} \tag{7.10}$$

where matrix $M_0$ is as defined in the Appendix. Note that all of the matrices $Q_j, \forall j \in \{1, 2\}$, $P, M_0, M_1, F$ and $L_g, \forall g \in \{1, \ldots, 10\}$ are sparse matrices.

We now establish the main result, i.e., that the pair of strategies $(u^{t^*}, \{v_1^{t^*}, v_2^{t^*}\})$ form a pair of Nash equilibrium. Note that if the optimal strategies of the trackers and the pursuer form a pair of Nash equilibrium, there is no incentive for the trackers to deviate from their optimal strategy (see Definition 72). This means that the pursuer has the correct belief of the trackers strategy and can determine its state-feedback strategy, $u^{t^*}$ by first solving the QCQP to determine $v_1^{t^*}$ and $v_2^{t^*}$ and then substituting these into (7.6). However, to determine the strategy of the trackers, the pursuer needs the information of the estimates that the trackers have of the target's location. Since the pursuer does not have this information, we propose that the pursuer uses the true value of the target's location to solve the QCQP and consequently determine $u^{t^*}$.

**Theorem 74.** *At every instant $t_e \leq t < t_f$, the pair of strategies $(u^{t^*}, \{v_1^{t^*}, v_2^{t^*}\})$ defined in* (7.6) *and obtained by solving the optimization problem* (7.8)*, form a pair of Nash equilibrium strategies for the payoff function $J(\hat{s}, e_i^t, p^t, v_i^t, u^t)$ as defined in* (7.3).

*Proof.* Observe that once the estimates about the location of the target converges to the true value, all of the mobile agents use the same value of the target's location to solve the QCQP.

Further, at every time instant $t_e \leq t < t_f$, the optimal strategy of the pursuer defined in equation (7.6) is the best-response of the pursuer to the trackers strategy. Similarly, the optimal strategy of the trackers obtained by solving the optimization problem defined (7.9) (if $\alpha = 1$) and (7.10) (otherwise) is the best response of the trackers to an optimal pursuer strategy. The result then follows directly from Definition 72. This concludes the proof. □

We now briefly describe how the game is solved. At every time instant, depending on the value of $\alpha$, the pursuer solves the optimization problem 7.9 or 7.10 using the true location of the target ($s$) to obtain $v_1^{t^*}$ and $v_2^{t^*}$. The pursuer then moves to the location by determining its control via 7.6. On the other hand, the trackers jointly solve the same optimization problem using the estimates of the target ($\hat{s}$) and move to the next location using $v_1^{t^*}$ and $v_2^{t^*}$.

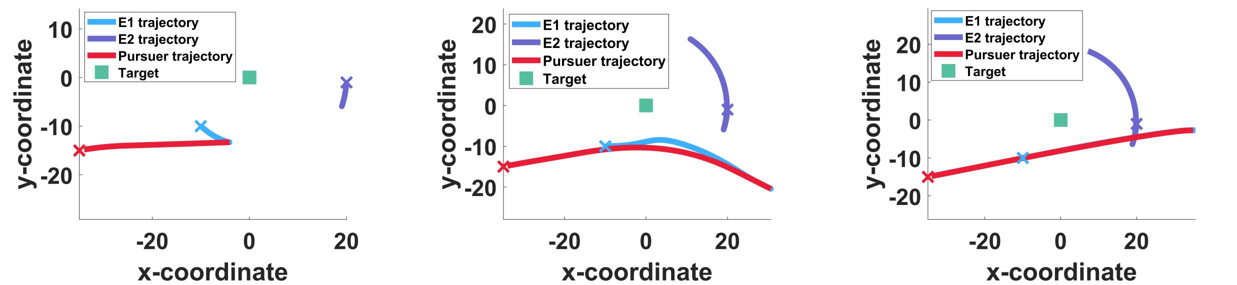
**Remark 13** (Bearing Measurements)**.** *If the trackers use a sensor that measures the bearing (angle) of the target relative to their positions instead of range measurements, then the determinant of the FIM is given by [60]*

$$f(\hat{s}, e_1^t, e_2^t, v_1^t, v_2^t) = \frac{1}{\sigma_V^2 \hat{S}_1^4 \hat{S}_2^4}\Big((\hat{X}_1^t - v_{x,1}^t)(\hat{Y}_2^t - v_{y,2}^t) - (\hat{Y}_1^t - v_{y,1}^t)(\hat{X}_2^t - v_{x,2}^t)\Big)^2.$$

*As the pursuer's optimal strategy does not change, by following similar steps as in Section 7.2.2, the optimization problem for the trackers can similarly be expressed as a QCQP and thus, this work easily extends to scenarios when trackers have access to bearing measurements.*

## 7.3 Numerical Observations

We now present numerical simulations of the optimal strategies defined in Section 7.2 and highlight the trajectories of the mobile agents. In all of our simulations, the parameter $\sigma_V^2$ was kept fixed to 0.03 and the target's location was chosen to be $(0, 0)$. Due to the number and size of the sparse matrices in the proposed QCQP optimization problem (7.9), generating the trajectories was time consuming. Thus, we use fmincon function in MATLAB to determine the optimal strategies of the trackers which was verified to be consistent with the strategies obtained by solving optimization problem in (7.9).

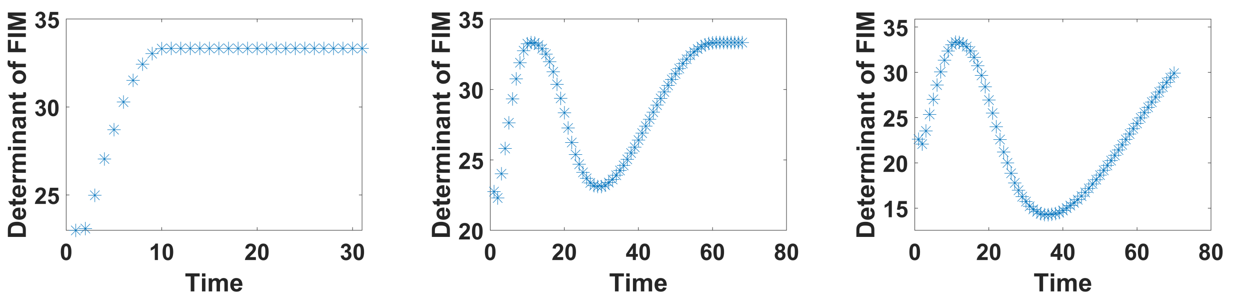



(a) Trajectories of the mobile agents for $\delta = 0$. $t_f = 44$.

(b) Trajectories of the mobile agents for $\delta = 0.2$. $t_f = 68$.

(c) Trajectories of the mobile agents for $\delta = 5$. $t_f = 71$.

Figure 7.3 Trajectories of the pursuer and the trackers for different values of $\delta$. The cross represents the starting locations of the mobile agents. The target is denoted by the green square.




(a) $\delta = 0$

(b) $\delta = 0.2$

(c) $\delta = 5$

Figure 7.4 Determinant of FIM vs Time plots for different values of $\delta$.

### 7.3.1 Example 1 ($\alpha = 1$)

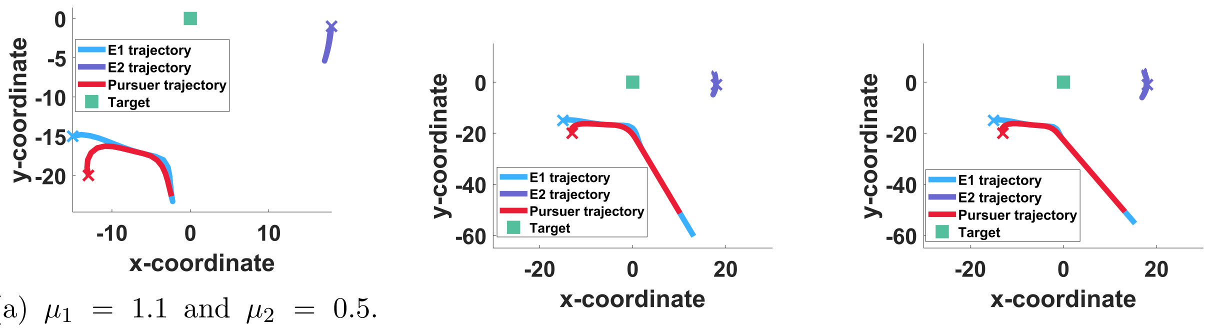
Our first numerical simulation (cf. Fig. 7.3) focuses on the trajectories of the mobile agents when the pursuer moves to capture the first tracker, $E_1$. Specifically, we select the initial locations such that $\alpha = 1$. To highlight the role of evasion by the trackers, we provide a numerical plot with $\delta = 0$ in Fig. 7.3, i.e., the evaders move to maximize only the determinant of the FIM. Note that the time taken by the pursuer to capture $E_1$ is mentioned in the description of each sub-figure in Fig. 7.3. The initial locations for all of the simulations presented in Fig. 7.3 were kept the same and selected to be $(-10, -10)$, $(20, -1)$ and $(-35, -15)$ for $E_1$, $E_2$ and the pursuer, respectively. Further, the parameters $\mu_1$ and $\mu_2$ were set to be 0.65 and 0.5, respectively.

Observe that in Fig. 7.3a, the trackers move to position themselves such that the angle subtended at the target by the position of the trackers is $\frac{\pi}{2}$. This is consistent with trajectories that maximize only the FIM as reported in [13]. Upon reaching that position, the trackers remain at that position until tracker $E_1$ is captured by the pursuer. Based on the value of $\delta$ in Fig. 7.3b as well as in Fig. 7.3c, observe that the tracker $E_1$ first moves away from the pursuer and then it moves away from the target, maximizing both the time to capture as well as the determinant of the FIM. Fig. 7.3b shows the cooperative behaviour of $E_2$. In particular, although the pursuer does not move towards $E_2$, tracker $E_2$ first moves downwards and then changes its direction in order maximize the determinant of the FIM by moving to a location such that the position of the trackers subtend an angle of $\frac{\pi}{2}$ at the target. Once the angle between the position of the trackers is $\frac{\pi}{2}$, tracker $E_2$ remains stationary at its location while $E_1$ evades

Finally, in Fig. 7.4, observe that the determinant of the FIM monotonically increases in Fig. 7.4a and then converges to 33.33. Although in Fig. 7.4b the determinant of FIM reaches the value 33.3, the value then decreases as the trackers cannot stay at that position due to the evasion cost. Note that at time $t = 60$, the cost converges to 33.3 highlighting the fact that the angle subtended by the position of the trackers to the target is now at $\frac{\pi}{2}$, and thus, tracker $E_2$ remains at its position whereas tracker $E_1$ moves in a straight line maintaining the same angle. Similar trend is observed in Fig. 7.4c. However, tracker $E_1$ is captured before the angle subtended by the trackers to the target is $\frac{\pi}{2}$. This is due to the higher value of $\delta$ as compared to that in Fig. 7.4b because of which tracker $E_1$ moves directly away from the pursuer. Thus, the trackers require more time to reach the positions from which the angle subtended to the target is $\frac{\pi}{2}$.

### 7.3.2 Example 2 (Faster trackers)

This numerical simulation considers a scenario that at one tracker is faster than the pursuer. The initial locations of the trackers and the pursuer was set to $(18, -1)$, $(-15, -15)$ and $(-13, -20)$, respectively. Finally the parameter $\delta$ was kept fixed to 0.14 and from the initial locations, $\alpha = 1$.




(a) $\mu_1 = 1.1$ and $\mu_2 = 0.5$. $t_f = 17$.

(b) $\mu_1 = 1.2$ and $\mu_2 = 0.5$.

(c) $\mu_1 = 1.1$ and $\mu_2 = 0.65$.

Figure 7.5 Trajectories of the pursuer and the trackers for $\delta = 0.14$ and different values of $\mu_i, \forall i \in \{1, 2\}$.

In Fig. 7.5a, although tracker $E_1$ is faster ($\mu_1 = 1.1$ and $\mu_2 = 0.5$) than the pursuer, the pursuer is able to capture tracker $E_1$. However, for the same initial locations of all of the mobile agents, the pursuer is unable to capture tracker $E_1$ when $\mu_1 = 1.2$ and $\mu_2 = 0.5$ (cf. Fig. 7.5b), implying that for faster trackers, there may exist *winning regions* for the pursuer as well as the trackers. Specifically, it may be possible to partition the environment into a winning region ($\Omega_P$) for the pursuer, i.e., the pursuer can always capture a tracker if the initial locations of all of the mobile agents lie inside $\Omega_P$. Similarly, it may be possible to characterize the winning region ($\Omega_T$) for the trackers, i.e., the trackers can always evade the pursuer if the initial locations of all of the mobile agents lie inside $\Omega_T$. Finally, observe that for the same initial locations and $\mu_1 = 1.1$ (cf. Fig. 7.5c), the pursuer cannot capture tracker $E_1$ if the speed of the tracker $E_2$ is increased from 0.5 (Fig. 7.5a) to 0.65.

We now describe how this work extends to two different scenarios. We start with a scenario with multiple targets followed by a scenario with multiple trackers.

## 7.4 Extensions

In this section, we describe how our analysis extends to the case of multiple targets and multiple trackers. We also show that in both scenarios the pursuer's optimal strategy remains the same as established in Section 7.2. We further establish that the optimization problem for the trackers can be converted to a QCQP.

### 7.4.1 Multiple targets

In this scenario, we consider that there are $N > 1$ targets, two mobile trackers and a single mobile pursuer. Each tracker has access only to range measurements from each of the $N$ targets. Thus, in this case, the measurement vector is $h(s_1, \ldots, s_N, e_1^t, e_2^t) = \begin{bmatrix} \|s_1 - e_1^t\| & \|s_1 - e_2^t\| & \ldots & \|s_N - e_1^t\| & \|s_N - e_2^t\| \end{bmatrix}'$, where $s_1, \ldots, s_N$ denote the fixed locations of the $N$ targets. By taking the partial derivatives with respect to the locations of the targets and replacing $s_j \forall j \in \{1, \ldots, N\}$ with its estimate $\hat{s}_j$, the FIM at time instant $t+1$ becomes a block diagonal matrix given by

$$F(\hat{s}_1, \ldots, \hat{s}_N, e_i^t, v_i^t) = \begin{bmatrix} f(\hat{s}_1, e_i^t, v_i^t) & \mathbf{0}_{2\times 2} & \ldots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & f(\hat{s}_2, e_i^t, v_i^t) & \ldots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \ldots & \ldots & f(\hat{s}_N, e_i^t, v_i^t) \end{bmatrix}$$

where $f(\hat{s}_j, e_i^t, v_i^t), \forall 1 \leq j \leq N$ is the FIM defined analogously as $f(\hat{s}_1, e_i^t, v_i^t)$ (see Section 3.1). Using the fact that determinant of a block diagonal matrix is the product of the determinant of its blocks yields

$$\det(F(\hat{s}_1, \ldots, \hat{s}_N, e_i^t, v_i^t)) = \prod_{j=1}^{N} \det(f(\hat{s}_j, e_i^t, v_i^t)).$$

Thus, the expression for the payoff is given by

$$\begin{aligned} J(\hat{s}_1, \ldots, \hat{s}_N, e_i^t, p^t, v_i^t, u^t) = {} & \det(F(\hat{s}_1, \ldots, \hat{s}_N, e_i^t, v_i^t)) + \\ & \delta(\alpha\|e_1^t + v_1^t - p^t - u^t\|^2 + (1-\alpha)\|e_2^t + v_2^t - p^t - u^t\|^2). \end{aligned}$$

Since the determinant of the FIM is not a function of the pursuer's control, it follows that the pursuer's strategy remains the same as defined in (7.6). Observe that $\det(F(\hat{s}_1, \ldots, \hat{s}_N, e_i^t, v_i^t))$ is a polynomial function of $v_{x,i}^t$ and $v_{y,i}^t$ for all $i \in \{1, 2\}$. Therefore, following similar steps as in Section 7.2 and from the fact that any polynomial can be expressed into the standard QCQP form [50], it follows that the optimization problem obtained for the trackers after substituting $u^{t^*}(v_1^t, v_2^t)$ can also be converted into a QCQP of the same form as defined in Lemma 73. Finally, given that the pair of strategies $(u^{t^*}, \{v_1^{t^*}, v_2^{t^*}\})$ are best responses to each other, it follows that the pair of strategies forms a Nash equilibrium.

### 7.4.2 Multiple trackers

We now consider the scenario with a single target, $M > 2$ trackers and a single mobile pursuer.

Let at time instant $t < t_f$, $\alpha \triangleq \begin{bmatrix} \alpha_1 & \dots & \alpha_M \end{bmatrix}' \in \mathbb{R}^M$ such that $\sum_{j=1}^{M} \alpha_j = 1$ and $\alpha_j \in \{0, 1\}, \forall j \in \{1, \dots, M\}$. Let $\mathbf{D} \in \mathbb{R}^M$ denote a vector consisting of the distance between the pursuer and the trackers, i.e., $\begin{bmatrix} \|p^t - e_1^t\| & \dots & \|p - e_M^t\| \end{bmatrix}'$. Then, the payoff is given by

$$J(\hat{s}, e_1^t, \dots, e_M^t, v_1^t, \dots, v_M^t, p^t) = \det(f(\hat{s}, e_1^t, \dots, e_M^t, v_1^t, \dots, v_M^t)) + \delta \alpha_t' \mathbf{D},$$

where

$$\det(f(\hat{s}, e_1^t, \dots, e_M^t, v_1^t, \dots, v_M^t)) = \frac{1}{\sigma_V^2} \sum_{j=1}^{M} \sum_{l=j+1}^{M} \frac{1}{\hat{S}_j^2 \hat{S}_l^2} \Big( (\hat{X}_j^t - v_{x,j}^t)(\hat{Y}_l^t - v_{y,l}^t) - (\hat{Y}_j^t - v_{y,j}^t)(\hat{X}_l^t - v_{x,l}^t) \Big)^2 .$$

For a given vector $\alpha$ at the first time instant, the strategy of the pursuer is the same as defined in Section 7.2 and thus, following similar steps, the payoff for the trackers can be expressed as a polynomial function in the optimization variables $v_{x,i}^t$ and $v_{y,i}^t$ for all $i \in \{1, \dots, M\}$. Hence, following similar steps as in Section 7.2 and given the fact that any polynomial can be expressed into the standard QCQP form [50], it follows that the optimization problem obtained for the trackers after substituting $u^{t^*}(v_1^t, v_2^t)$ can also be converted into a QCQP of the same form as defined in Lemma 73.

Finally, given that the pair of strategies $(u^{t^*}, \{v_1^{t^*}, \dots, v_M^{t^*}\})$ are best responses to each other, it follows that the pair of strategies forms a Nash equilibrium.

## 7.5 Conclusions

This chapter introduced a tracking-evasion game consisting of a single pursuer, two trackers and a single target. The pursuer seeks to deter the tracking performance of the trackers by minimizing the square of the distance to the closest tracker, whereas, the trackers aim to jointly maximize a weighted combination of the determinant of the Fisher Information Matrix and the square of the distance between the pursuer to the tracker being pursued. We determined optimal strategies of the pursuer and and showed that the optimal strategies of the trackers can be obtained by solving a Quadratically Constrained Quadratic Program. We then established that the pair of strategies form a Nash equilibrium and provided several numerical observations highlighting the trajectories and the payoff. Finally, we discussed the extension of this work to multiple trackers and multiple targets.

# CHAPTER 8

# CONCLUSIONS AND FUTURE DIRECTIONS

In this work, we considered perimeter defense problems in linear and conical environments consisting of 1) a single defender, 2) multiple homogeneous defenders, and 3) heterogeneous defenders and designed online algorithms with finite competitive ratios and established necessary conditions for the existence of algorithms with finite competitive ratios. We also addressed two surveilling problems. Specifically, we first address a joint motion planning for a Dubins-Laser system which consists of a laser attached to a Dubins vehicle. Apart from establishing multiple properties of the shortest path including the cooperative nature of the Dubins vehicle and the laser, we establish that the shortest path is $\{CSC, CC\}$ or a subsegment of it. Second, we consider a scenario in which intruders need to estimate the location of a perimeter while the defenders move to deter the intruders from obtaining the measurements and determined Nash equilibrium strategies for both the defenders and the intruders.

There are numerous possible extensions of these works, including defenders with sensing constraints or perimeter defense in higher dimensional environments. We outline some of the extensions of this work for future direction in the next two subsections.

### 8.0.1 Competitive Perimeter Defense Problems

An immediate future direction for our perimeter defense works (Chapters 2-5) is to close the gap between the established fundamental limits and the online algorithms. Another future direction is to consider the perimeter defense problems in graphical environments with single or multiple defenders. In Chapter 2 and Chapter 3, we addressed perimeter defense for linear environments which can serve as a starting point for general graphs. Investigating how the competitive ratio of online algorithms vary with the properties of various graphical environments, such as the degree and the diameter of the graph will provide valuable insights. One thing to note is that a graphical environment can have multiple nodes to defend which may lead to completely different algorithms than the ones designed in this work.

Another possible direction is to explore learning based approaches for these problems. Reinforcement learning is a valuable tool that can be used to design algorithms for the defenders that learn to maximize the number of intruders captured over time. Similar to competitive analysis, reinforcement learning provides a notion of *regret*, i.e., the difference between the performance of a static optimal offline algorithm and an online algorithm. This further leads to the question of designing algorithms that minimize regret as well as competitive ratios.

### 8.0.2 Surveillance Problems

For the Dubins-Laser System, there are numerous future directions. One such direction is to consider multiple static targets in the environment that the Dubins-Laser System need to sequentially capture. Finding the shortest tour that require a Dubins vehicle to visit multiple static locations are known as Dubins TSP or D-TSP problems and are known to be NP-Hard [58]. We believe that a similar result can be established for the Dubins-Laser System as well. Designing approximation algorithms as well as bounds on the shortest path or designing online algorithms with finite competitive ratios are possible ways to approach the Dubins-Laser System TSP problem.

Another future direction of these works is to consider a single or multiple moving intruders that the Dubins-Laser System must capture. The intruders may move on a known trajectory or may move adversarially to evade the vehicle. Another possible related future direction is when the Dubins-Laser System needs to move to keep a moving intruder within the range of the laser for some finite amount of time. Simultaneous capture of intruders via multiple such defenders or multiple defenders and multiple intruders are also possible future directions.

Finally, for Chapter 7, apart from leveraging the sparse-structure of the matrices for the optimization problem, a key future direction includes a generalized setup with multiple pursuers and trackers with motion and energy constraints.

## BIBLIOGRAPHY


[1] Z. Akilan and Z. Fuchs. Zero-sum turret defense differential game with singular surfaces. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 2041–2048, 2017.

[2] Amna AlDahak and Ashraf Elnagar. A practical pursuit-evasion algorithm: Detection and tracking. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 343–348. IEEE, 2007.

[3] Shivam Bajaj and Shaunak D Bopardikar. Dynamic boundary guarding against radially incoming targets. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4804–4809. IEEE, 2019.

[4] Shivam Bajaj and Shaunak D Bopardikar. Optimal pursuit of surveilling agents near a high value target. In *International Conference on Decision and Game Theory for Security*, pages 168–187. Springer, 2022.

[5] Shivam Bajaj, Shaunak D Bopardikar, Alexander Von Moll, Eric Torng, and David W Casbeer. Competitive perimeter defense with a turret and a mobile vehicle. *Frontiers in Control Engineering*, 4:1128597, 2023.

[6] Shivam Bajaj, Shaunak D. Bopardikar, Eric Torng, Alexander Von Moll, and David W. Casbeer. Competitive perimeter defense with multiple vehicles. *IEEE Transactions on Robotics*, 2022. Under Review.

[7] Shivam Bajaj, Shaunak D Bopardikar, Alexander Von Moll, Eric Torng, and David W Casbeer. Perimeter defense using a turret with finite range and startup time. In *2023 American Control Conference (ACC)*, pages 3350–3355. IEEE, 2023.

[8] Shivam Bajaj, Eric Torng, and Shaunak D Bopardikar. Competitive perimeter defense in linear environments. *Theoretical Computer Science*. Under Review. Online available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4145446.

[9] Shivam Bajaj, Eric Torng, and Shaunak D. Bopardikar. Competitive perimeter defense on a line. In *2021 American Control Conference (ACC)*, pages 3196–3201, 2021.

[10] Shivam Bajaj, Eric Torng, Shaunak D Bopardikar, Alexander Von Moll, Isaac Weintraub, Eloy Garcia, and David W Casbeer. Competitive perimeter defense of conical environments. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6586–6593. IEEE, 2022.

[11] Reuven Bar-Yehuda, Guy Even, and Shimon Moni Shahar. On approximating a geometric prize-collecting traveling salesman problem with time windows. *Journal of Algorithms*, 55(1):76–92, 2005.

[12] Dimitris J Bertsimas and Garrett Van Ryzin. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Operations Research*, 39(4):601–615, 1991.

[13] Adrian N Bishop, Barış Fidan, Brian DO Anderson, Kutluyıl Doğançay, and Pubudu N Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[14] François Blais. Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1):231–243, 2004.

[15] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. Shortest paths of bounded curvature in the plane. *Journal of Intelligent and Robotic Systems*, 11:5–20, 1994.

[16] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 2005.

[17] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] Xuân-Nam Bui, J-D Boissonnat, Philippe Soueres, and J-P Laumond. Shortest path synthesis for dubins non-holonomic robot. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2–7. IEEE, 1994.

[19] Xuân-Nam Bui and Jean-Daniel Boissonnat. *Accessibility region for a car that only moves forwards along optimal paths*. PhD thesis, INRIA, 1994.

[20] Francesco Bullo, Emilio Frazzoli, Marco Pavone, Ketan Savla, and Stephen L Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504, 2011.

[21] Alessandra Capolupo, Stefania Pindozzi, Collins Okello, Nunzio Fiorentino, and Lorenzo Boccia. Photogrammetry for environmental monitoring: The use of drones and hydrological models for detection of soil contaminated by copper. *Science of the Total Environment*, 514:298–306, 2015.

[22] M Casey. Drones bring fight and flight to battle against poachers. *Scientific American (online)*, 20, 2014.

[23] Chaw-Bing Chang and John Tabaczynski. Application of state estimation to target tracking. *IEEE Transactions on automatic control*, 29(2):98–109, 1984.

[24] Mo Chen, Zhengyuan Zhou, and Claire J Tomlin. Multiplayer reach-avoid games via pairwise outcomes. *IEEE Transactions on Automatic Control*, 62(3):1451–1457, 2016.

[25] Timothy H Chung, Joel W Burdick, and Richard M Murray. A decentralized motion coordination strategy for dynamic target tracking. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2416–2422. IEEE, 2006.

[26] Demetris Coleman, Shaunak D Bopardikar, and Xiaobo Tan. Observability-aware target tracking with range only measurement. In *2021 American Control Conference (ACC)*, pages 4217–4224. IEEE, 2021.

[27] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957.

[28] Jacob T English and Jay P Wilhelm. Defender-aware attacking guidance policy for the target–attacker–defender differential game. *Journal of Aerospace Information Systems*, 18(6):366–376, 2021.

[29] U.S. Air Force. U.S. Air Force 2030 Science and Technology Strategy. Technical report, 2019.

[30] Roee Mordechai Francos and Alfred M Bruckstein. Search for smart evaders with swarms of sweeping agents. *IEEE Transactions on Robotics*, 2021.

[31] Jie Gao, Su Jia, Joseph SB Mitchell, and Lu Zhao. Approximation algorithms for time-window tsp and prize collecting tsp problems. In *Algorithmic Foundations of Robotics XII*, pages 560–575. Springer, 2020.

[32] Eloy Garcia, David W Casbeer, and Meir Pachter. Active target defence differential game: fast defender case. *IET Control Theory & Applications*, 11(17):2985–2993, 2017.

[33] Eloy Garcia, David W Casbeer, and Meir Pachter. Optimal target capture strategies in the target-attacker-defender differential game. In *2018 Annual American Control Conference (ACC)*, pages 68–73. IEEE, 2018.

[34] Eloy Garcia, David W Casbeer, and Meir Pachter. Optimal strategies for a class of multi-player reach-avoid differential games in 3d space. *IEEE Robotics and Automation Letters*, 5(3):4257–4264, 2020.

[35] Eloy Garcia, David W Casbeer, Alexander Von Moll, and Meir Pachter. Multiple pursuer multiple evader differential games. *IEEE Transactions on Automatic Control*, 66(5):2345–2350, 2020.

[36] Eloy Garcia, Alexander Von Moll, David W Casbeer, and Meir Pachter. Strategies for defending a coastline against multiple attackers. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7319–7324. IEEE, 2019.

[37] Luis Guerrero-Bonilla, Carlos Nieto-Granda, and Magnus Egerstedt. Robust perimeter defense using control barrier functions. In *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 11 2021.

[38] Sandra Gutiérrez, Sven O Krumke, Nicole Megow, and Tjark Vredeveld. How to whack moles. *Theoretical computer science*, 361(2-3):329–341, 2006.

[39] João P Hespanha. *Noncooperative game theory: An introduction for engineers and computer scientists*. Princeton University Press, 2017.

[40] Nguyen T Hung, Francisco FC Rego, and António M Pascoal. Cooperative distributed estimation and control of multiple autonomous vehicles for range-based underwater target localization and pursuit. *IEEE Transactions on Control Systems Technology*, 2021.

[41] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization.* Courier Corporation, 1999.

[42] Beaubien Jason. In the russia-ukraine war, drones are one of the most powerful weapons. *NPR*, 2022.

[43] Kristofer S Kappel, Tauã M Cabreira, João L Marins, Lisane B de Brisolara, and Paulo R Ferreira. Strategies for patrolling missions with multiple uavs. *Journal of Intelligent & Robotic Systems*, 99(3):499–515, 2020.

[44] C Yalçın Kaya. Markov–dubins path via optimal control theory. *Computational Optimization and Applications*, 68(3):719–747, 2017.

[45] Sven Krumke. *Online optimization: Competitive analysis and beyond.* PhD thesis, 2002.

[46] Elijah S Lee, Daigo Shishika, and Vijay Kumar. Perimeter-defense game between aerial defender and ground intruder. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1530–1536. IEEE, 2020.

[47] Li Liang, Fang Deng, Zhihong Peng, Xinxing Li, and Wenzhong Zha. A differential game for cooperative target defense. *Automatica*, 102:58–71, 2019.

[48] Joseph Lorenzetti, Mo Chen, Benoit Landry, and Marco Pavone. Reach-avoid games via mixed-integer second-order cone programming. In *2018 IEEE conference on decision and control (CDC)*, pages 4409–4416. IEEE, 2018.

[49] Douglas G Macharet, Austin K Chen, Daigo Shishika, George J Pappas, and Vijay Kumar. Adaptive partitioning for coordinated multi-agent perimeter defense. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7971–7977. IEEE, 2020.

[50] Ramtin Madani, Ghazal Fazelnia, and Javad Lavaei. Rank-2 matrix solution for semidefinite relaxations of arbitrary polynomial optimization problems. *constraints*, 21:25, 2014.

[51] Kostas Margellos and John Lygeros. Hamilton–Jacobi formulation for reach–avoid differential games. *IEEE Transactions on Automatic Control*, 56(8):1849–1861, 2011.

[52] Sonia Martínez and Francesco Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[53] M Anwar Ma'Sum, M Kholid Arrofi, Grafika Jati, Futuhal Arifin, M Nanda Kurniawan, Petrus Mursanto, and Wisnu Jatmiko. Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance. In *2013 international conference on advanced computer science and information systems (ICACSIS)*, pages 161–166. IEEE, 2013.

[54] Timothy G McGee and J Karl Hedrick. Guaranteed strategies to search for mobile evaders in the plane. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[55] AB Miller and BM Miller. Underwater target tracking using bearing-only measurements. *Journal of Communications Technology and Electronics*, 63(6):643–649, 2018.

[56] Douglas Moura Miranda and Samuel Vieira Conceição. The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, 64:104–116, 2016.

[57] Rafael Murrieta-Cid, Ubaldo Ruiz, Jose Luis Marroquin, Jean-Paul Laumond, and Seth Hutchinson. Tracking an omnidirectional evader with a differential drive robot. *Autonomous Robots*, 31(4):345–366, 2011.

[58] Jerome Ny, Eric Feron, and Emilio Frazzoli. On the dubins traveling salesman problem. *IEEE Transactions on Automatic Control*, 57(1):265–270, 2011.

[59] Marco Pavone, Nabhendra Bisnik, Emilio Frazzoli, and Volkan Isler. A stochastic and dynamic vehicle routing problem with time windows and customer impatience. *Mobile Networks and Applications*, 14(3):350–364, 2009.

[60] Sameera Ponda, Richard Kolacinski, and Emilio Frazzoli. Trajectory optimization for target localization using small unmanned aerial vehicles. In *AIAA guidance, navigation, and control conference*, page 6015, 2009.

[61] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC press, 1987.

[62] Arman Pourghorban and Dipankar Maity. Target defense with sequentially arriving attackers. In *2022 Conference on Decision and Control(CDC)*. IEEE, 2022.

[63] Arman Pourghorban and Dipankar Maity. Target defense against periodically arriving intruders. In *2023 American Control Conference (ACC)*, pages 1673–1679, 2023.

[64] Harilaos N Psaraftis. Dynamic vehicle routing problems. *Vehicle routing: Methods and studies*, 16:223–248, 1988.

[65] Jake D Quenzer and Kristi A Morgansen. Observability based control in range-only underwater vehicle localization. In *2014 American control conference*, pages 4702–4707. IEEE, 2014.

[66] Mohammadhussein Rafieisakhaei, Suman Chakravorty, and P. R. Kumar. On the use of the observability gramian for partially observed robotic path planning problems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1523–1528, 2017.

[67] Andrey V Savkin and Hailong Huang. A method for optimized deployment of a network of surveillance aerial drones. *IEEE Systems Journal*, 13(4):4474–4477, 2019.

[68] Daigo Shishika and Vijay Kumar. A review of multi agent perimeter defense games. In *International Conference on Decision and Game Theory for Security*, pages 472–485. Springer, 2020.

[69] Daniel D Sleator and Robert E Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[70] Stephen L Smith, Shaunak D Bopardikar, and Francesco Bullo. A dynamic boundary guarding problem with translating targets. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8543–8548. IEEE, 2009.

[71] Alexander Solodov, Adam Williams, Sara Al Hanaei, and Braden Goddard. Analyzing the threat of unmanned aerial vehicles (uav) to nuclear facilities. *Security Journal*, 31(1):305–324, 2018.

[72] John R Spletzer and Camillo J Taylor. Dynamic sensor planning and control for optimally tracking targets. *The International Journal of Robotics Research*, 22(1):7–20, 2003.

[73] Domagoj Tolić and Rafael Fierro. Adaptive sampling for tracking in pursuit-evasion games. In *2011 IEEE International Symposium on Intelligent Control*, pages 179–184. IEEE, 2011.

[74] Athanasios Tsoukalas, Daitao Xing, Nikolaos Evangeliou, Nikolaos Giakoumidis, and Anthony Tzes. Deep learning assisted visual tracking of evader-uav. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 252–257. IEEE, 2021.

[75] Shridhar Velhal, Suresh Sundaram, and Narasimhan Sundararajan. A decentralized multirobot spatiotemporal multitask assignment approach for perimeter defense. *IEEE Transactions on Robotics*, 2022.

[76] Shridhar B Velhal, Suresh Sundaram, et al. Dynamic resource allocation with decentralized multi-task assignments (dream) approach for perimeter defense problem. *IEEE Transactions on Aerospace and Electronic Systems*, 2022.

[77] Alexander Von Moll and Zachariah Fuchs. Optimal constrained retreat within the turret defense differential game. In *Conference on Control Technology and Applications*, 2020.

[78] Alexander Von Moll and Zachariah Fuchs. Turret lock-on in an engage or retreat game. In *American Control Conference*, pages 3188–3195. IEEE, 2021.

[79] Alexander Von Moll, Meir Pachter, Daigo Shishika, and Zachariah Fuchs. Circular target defense differential games. *Transactions on Automatic Control*, 68, 2022.

[80] Alexander Von Moll, Daigo Shishika, Zachariah Fuchs, and Michael Dorothy. The turret-runner-penetrator differential game with role selection. *Transactions on Aerospace & Electronic Systems*, 2022.

[81] Rui Yan, Zongying Shi, and Yisheng Zhong. Reach-avoid games with two defenders and one attacker: An analytical approach. *IEEE transactions on cybernetics*, 49(3):1035–1046, 2018.

[82] Rui Yan, Zongying Shi, and Yisheng Zhong. Task assignment for multiplayer reach–avoid games in convex domains via analytical barriers. *IEEE Transactions on Robotics*, 36:107–124, 2019.

[83] Chun Yang, Lance Kaplan, and Erik Blasch. Performance measures of covariance and information matrices in resource management for target state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2594–2613, 2012.

[84] Jing Zhang and Jun Zhuang. Modeling a multi-target attacker-defender game with multiple attack types. *Reliability Engineering & System Safety*, 185:465–475, 2019.

[85] Ke Zhou and Stergios I Roumeliotis. Optimal motion strategies for range-only constrained multisensor target tracking. *IEEE Transactions on Robotics*, 24(5):1168–1185, 2008.

[86] Ke Zhou and Stergios I Roumeliotis. Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, 27(4):678–695, 2011.

[87] Rui Zou and Sourabh Bhattacharya. On optimal pursuit trajectories for visibility-based target-tracking game. *IEEE Transactions on Robotics*, 35(2):449–465, 2018.

## APPENDIX A: PROOFS

We now provide proofs for Lemma 41 and Lemma 42 established for Algorithm Dec-SNP and proofs for Lemma 47 and Lemma 48 for Algorithm Coop-SNP.

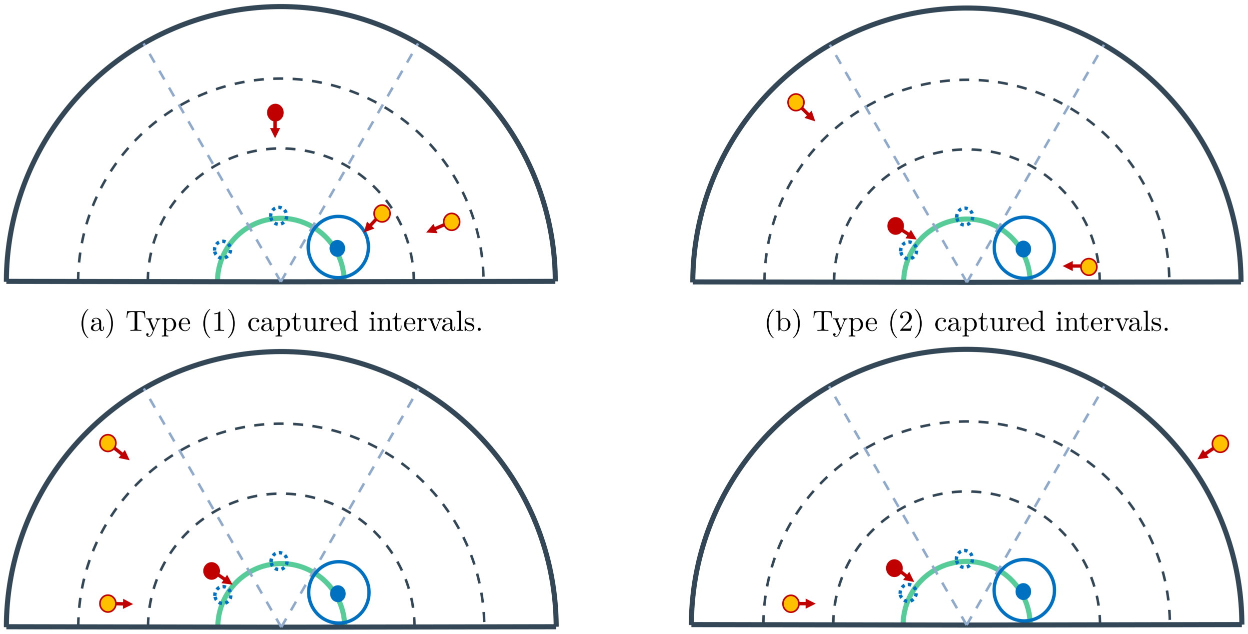
### Dec-SNP Algorithm

For the proof of Lemma 41 and as a consequence of Lemma 40, we observe that any two consecutive captured intervals of defender $V_m, 1 \leq m \leq M$, assumed be located at the resting point of sector $N_{i,m}$, can be classified into four types;

1. stay at the current location and capture the first two intervals in sector $N_{i,m}$ (cf. Figure 1a).
2. stay at the current location and capture the first interval in $N_{i,m}$ and then move to the resting point of sector $N_{o,m}$ and capture the third interval (cf. Figure 1b), where $N_{o,m}$ is the sector that is chosen after **C1**.
3. move to the resting point of $N_{o,m}$ and capture both (second and third) intervals (cf. Figure 1c).
4. move to the resting point of sector $N_{o,m}$ and capture the second interval and then move to the resting point of another sector, $N_{o',m}$, to capture its second interval (cf. Figure 1d). Here, sector $N_{o',m}, o' \in \{1, \ldots, n_s\} \setminus \{o\}$, denotes the sector chosen after **C1** once the defender finishes capturing an interval from the sector $N_{o,m}$.

*Proof of Lemma 41.* The aim of this proof is to establish that for each of the four types of captured intervals identified above, the defender $V_m$, for any $m \in \{1, \ldots, n_s\}$, is charged $3(n_s - 1)$ times. We first consider type (1) captured intervals $S_{i,m}^{j+1}$ and $S_{i,m}^{j+2}$.

Since defender $V_m$ decides at time instant $jD$ and $(j+1)D$ to capture $S_{i,m}^{j+1}$ and $S_{i,m}^{j+2}$, respectively, it loses $S_{l,m}^{j+2}$ and $S_{l,m}^{j+3}$, respectively, from the other remaining sectors of partition $m$, i.e., $l \in \{1, \ldots, n_s\} \setminus \{i\}$. Thus, the two captured intervals are charged $2n_s - 2$ times. The remaining $n_s - 1$ charge is as follows. Since the defender $V_m$ is currently located at



(a) Type (1) captured intervals.

(b) Type (2) captured intervals.

(c) Type (3) captured intervals.

(d) Type (4) captured intervals.

Figure 1 Description of captured intervals for Coop-SNP (resp. Dec-SNP) for a single defender located at the resting point of sector $\delta_3$ (resp. $N_{3,m}$). The captured intervals are highlighted with intruders in yellow color.

$(x_i^m, \alpha_i^m)$, it follows that the defender captured $S_{i,m}^j$. This implies that the comparison **C1** must have yielded sector $N_{i,m}$ at either time instant $(j-2)D$ (if the defender was located at $(x_l^m, \alpha_l^m), l \neq i)$) or $(j-1)D$ (if the defender was located at $(x_i^m, \alpha_i^m)$. Recall that **C1** requires at least $S_{i,m}^j$ and $S_{i,m}^{j+1}$ for comparison. Since, defender $V_m$ captured $S_{i,m}^j$ followed by $S_{i,m}^{j+1}$, the captured intervals are charged $n_s - 1$ times for all $S_{l,m}^{j+1}$ intervals. Thus, the total charge is $3(n_s - 1)$.

Analogously, type (2) captured intervals $S_{i,m}^{j+1}$ and $S_{o,m}^{j+3}$ are also charged $n_s - 1$ times for lost intervals $S_{l,m}^{j+1}, l \in \{1, \ldots, n_s\} \setminus \{i\}$ and another $n_s - 1$ times for lost intervals $S_{l,m}^{j+2}$. The remaining $n_s - 1$ charge is as follows. One charge for all of the intervals $S_{i,m}^{j+2}, S_{i,m}^{j+3}$ and $S_{i,m}^{j+4}$ combined and $n_s - 2$ charge for $S_{l',m}^{j+3}$ and $S_{l',m}^{j+4}$, $l' \in \{1, \ldots, n_s\} \setminus \{i, o\}$, combined.

We now consider type (3) captured intervals. Type (3) captured intervals $S_{o,m}^{j+2}$ and $S_{o,m}^{j+3}$ are charged once for the lost intervals $S_{i,m}^{j+1}, S_{i,m}^{j+2}$ and $S_{i,m}^{j+3}$ combined and $n_s - 2$ times for the lost intervals $S_{l,m}^{j+2}$ and $S_{l,m}^{j+3}$, $l \in \{1, n_s\} \setminus \{i, o\}$. The captured intervals are charged another $n_s - 1$ times for the lost intervals $S_{l,m}^{j+4}$, $l \in \{1, \ldots, n_s\} \setminus \{o\}$. Finally, the last $n_s - 1$ charge is for the lost intervals $S^{j}_{l,m}$ and $S^{j+1}_{l,m}, l \in \{1, \ldots, n_s\} \setminus \{i\}$ as defender $V_m$ captured $S^{j+2}_{o,m}$ and not $S^{j+1}_{i,m}$.

Finally, we consider type (4) captured intervals $S^{j+2}_{o,m}$ and $S^{j+4}_{o',m}$. For ease of presentation and without loss of generality, we assume $o' = i$. In other words, the defender moves from sector $N_{i,m}$ to sector $N_{o,m}$ to capture $S^{j+2}_{o,m}$. Then, defender moves back to sector $N_{i,m}$ to capture $S^{j+4}_{i,m}$. Type (4) captured intervals are charged once for the lost intervals $S^{j+1}_{i,m}, S^{j+2}_{i,m}$ and $S^{j+3}_{i,m}$ combined and $n_s - 2$ times for the lost intervals $S^{j+2}_{l,m}$ and $S^{j+3}_{l,m}$, $l \in \{1, n_s\} \setminus \{i, o\}$. Similarly, captured intervals are charged once for the lost intervals $S^{j+3}_{o,m}, S^{j+4}_{o,m}$ and $S^{j+5}_{o,m}$ combined and $n_s - 2$ times for the lost intervals $S^{j+4}_{l,m}$ and $S^{j+5}_{l,m}$, $l \in \{1, \ldots, n_s\} \setminus \{i, o\}$. Finally, the last $n_s - 1$ charge is for the lost intervals $S^{j}_{l,m}$ and $S^{j+1}_{l,m}, l \in \{1, \ldots, n_s\} \setminus \{i\}$ as defender $V_m$ captured $S^{j+2}_{o,m}$ and not $S^{j+1}_{i,m}$.

Since each type of captured intervals are charged $3n_s - 3$ times, and this result holds for every defender $V_m$, the result is established. □

*Proof for Lemma 42.* Since Dec-SNP directs every defender $V_m$ to stay at a resting point of any sector, in a partition $m$, for some time interval, it can be viewed as a sequence of *traces*, in which the defender $V_m$ spends some number of intervals at one resting point and some number of intervals at another. Each trace, of a partition $m$, is defined by a set $\{k_1, k_2, \ldots, k_{n_s}\}$, where each element $k_l$, $l \in \{1, \ldots, n_s\}$ denotes the number of intervals that the defender $V_m$ decides to capture by staying at the corresponding resting point of the sector $N_{l,m}$.

Note that for any partition $m$, any realization of Dec-SNP can be achieved by the combination of one or more traces as described in the following cases. Case (i) $k_i = 3$ and $k_l = 0 \ \forall l \in \{1, \ldots, n_s\} \setminus \{i\}$, Case (ii) $0 \leq k_i < 3$ and $k_o = 2$ and Case (iii) $k_i = 0$, $k_o = 1$ and $k_{o'} = 1, \forall o \in \{1, \ldots, n_s\} \setminus \{i\}$ and $\forall o' \in \{1, \ldots, n_s\} \setminus \{o\}$. The idea is to identify all of the lost and captured intervals in each case and show that each lost interval is accounted by the captured intervals.

**Case (i)**: Due to comparison steps **C1** and **C2** at time $jD$, the captured intervals $S^{j+1}_{i,m}$,

$S_{i,m}^{j+2}$ and $S_{i,m}^{j+3}$ account for all of the lost intervals $S_{l,m}^{j+2}$ and $S_{l,m}^{j+3}$, $\forall l \in \{1, \ldots, n_s\} \setminus \{i\}$. There are two sub-cases; sub-case (a) $N_o = N_i$ at time instant $jD$ and sub-case (b), there exists a sector $N_o \neq N_i$ at time instant $jD$ (comparison **C1**) such that $|S_{o,m}^{j+2}| < |S_{i,m}^{j+1}|$ (comparison **C2**). We first consider sub-case (a). Sub-case (a) implies that at time instant $jD$, the total number of intruders in sector $N_{i,m}$ is more than in any other sector in the environment. Thus, captured intervals $S_{i,m}^{j+1}$, $S_{i,m}^{j+2}$ and $S_{i,m}^{j+3}$ account for all of the lost intervals $S_{l,m}^{j+2}$ and $S_{l,m}^{j+3}$, $\forall l \neq i$. In sub-case (b), we account for lost intervals $S_{l,m}^{j+2}$, $S_{l,m}^{j+3}$, $\forall l \in \{1, \ldots, n_s\} \setminus \{i, o\}$ and $S_{o,m}^{j+2}$, $S_{o}^{j+3}$, separately. Lost intervals $S_{l,m}^{j+2}$ and $S_{l,m}^{j+3}$ are accounted for because $|S_{l,m}^{j+2}| + |S_{l,m}^{j+3}| \leq |S_{i,m}^{j+1}| + |S_{i,m}^{j+2}| + |S_{i,m}^{j+3}|$ or equivalently $\eta_i^l(m) \leq \eta_i^i(m)$ (comparison **C1**). Now it remains to account for lost intervals $S_{o,m}^{j+2}$ and $S_{o,m}^{j+3}$. Observe that if there exists a sector $N_{o,m} \neq N_i$ at time instant $jD$ such that $|S_{o,m}^{j+2}| < |S_{i,m}^{j+1}|$, then there cannot exist the same $N_{o,m}$ at time instant $(j+1)D$ (from comparison **C1**). Thus, even if $N_{o,m} \neq N_{i,m}$ exists, then the lost interval $S_{o,m}^{j+2}$ is accounted by $S_{i,m}^{j+1}$ as $|S_{o,m}^{j+2}| < |S_{i}^{j+1}|$ (comparison **C2**). Since, at time $(j+1)D$, sector $N_{o,m}$ cannot be selected again, it follows that $\eta_i^o(m) < \eta_i^i(m)$ at time $(j+1)D$ and thus, $S_{o,m}^{j+3}$ is accounted for.

**Case (ii)**: To account for the lost intervals $S_{l,m}^{j+k_i}$ and $S_{l,m}^{j+1+k_i}, \forall l \in \{1, \ldots, n_s\} \setminus \{i\}$, from comparison **C1** and **C2** at time $(j+k_i)D$, the defender was supposed to capture all $S_{i,m}^{j-2+k_i}$, $S_{i,m}^{j-1+k_i}, \ldots,$ $S_{i,m}^{j+1+k_i}$ intervals. While the defender captured $S_{i,m}^{j-2+k_i}, \ldots, S_{i,m}^{j+k_i}$ intervals, it did not capture $S_{i,m}^{j+1+k_i}$. As $\eta_i^o(m) > \eta_i^l(m)$ at time instant $(j+k_i)D$, lost intervals $S_{l,m}^{j+k_i}$ and $S_{l,m}^{j+1+k_i}, \forall l \in \{1, \ldots, n_s\} \setminus \{i\}$ are fully accounted for. The remaining lost intervals $S_{i,m}^{j+1+k_i}$, $S_{i,m}^{j+2+k_i}$, $S_{i,m}^{j+3+k_i}$ $S_{l,m}^{j+2+k_i}$, and $S_{l,m}^{j+3+k_i}$ $\forall l \in \{1, \ldots, n_s\} \setminus \{o\}$ are fully accounted by the captured intervals $S_{o,m}^{j+2+k_i}$ and $S_{o,m}^{j+3+k_i}$ because the conditions $\eta_i^o(m) > \eta_i^i(m)$ and $\eta_i^o(m) > \eta_i^l(m)$ are satisfied at time instant $(j+k_i)D$ (comparison **C1**).

**Case (iii)**: To account for lost intervals $S_{i,m}^{j+1}$, $S_{i,m}^{j+2}$, $S_{i,m}^{j+3}$, $S_{l,m}^{j+2}$, and $S_{l,m}^{j+3}$ $\forall l \in \{1, \ldots, n_s\} \setminus \{i, o\}$, the defender was supposed to capture $S_{o,m}^{j+2}$ and $S_{o,m}^{j+3}$. This follows because at time instant $jD$, $\eta_i^o(m) > \eta_i^i(m)$ (comparison **C1**) and $|S_{o,m}^{j+2}| \geq |S_{i,m}^{j+1}|$ (comparison **C2**). The defender captured $S_{o,m}^{j+2}$ which accounts for $S_{i,m}^{j+1}$ as $|S_{o,m}^{j+2}| \geq |S_{i,m}^{j+1}|$. As the defender moved to capture $S_{o',m}^{j+4}$ at time $(j+2)D$, it implies that $|S_{o',m}^{j+4}| \geq |S_{o,m}^{j+3}|$ (comparison **C2**) and thus, $S_{o,m}^{j+3}$, $S_{i,m}^{j+2}$, $S_{i,m}^{j+3}$, $S_{l,m}^{j+2}$, and $S_{l,m}^{j+3}$ are all accounted by the captured interval $|S_{o',m}^{j+4}|$. Finally, the lost intervals $|S_{l,m}^{j+4}|, \forall l \in \{1, \ldots, n_s\} \setminus \{o'\}$ are accounted for as follows: If the defender also captures $S_{o',m}^{j+5}$, then lost intervals $S_{l,m}^{j+4}$ are accounted for by per case (ii) ($k_i = 1$). Otherwise (i.e., the defender moved to another sector $N_{\tilde{o},m}, \tilde{o} \neq o$ to capture $S_{\tilde{o},m}^{j+6}$), $S_{l,m}^{j+4}$ is accounted for as per case (iii) as now the lost intervals will be $S_{i,m}^{j+3}$, $S_{i,m}^{j+4}$, $S_{i,m}^{j+5}$, $S_{l,m}^{j+4}$, and $S_{l,m}^{j+5}$ $\forall l \in \{1, \ldots, n_s\} \setminus \{i, o\}$.

Finally, note that the boundary cases of the first and the last intervals fall into these cases by adding dummy intervals $S_{i,m}^{0}, \forall i \in \{1, \ldots, n_s\}$ and $S_{i,m}^{Y+1}$, where $Y$ denotes the last interval that consists of intruders in any sector. We assume that the defender captures all of the dummy intervals. This concludes the proof. □

**Coop-SNP Algorithm**

Similar to that of Algorithm Dec-SNP, any two consecutive captured intervals of any defender, say $V_i \in \mathbf{V}$ and assumed be located at the resting point of sector $\delta_i$, can be classified into four types;

1. stay at the current location and capture the first two intervals in sector $\delta_i \in \tilde{\mathbf{O}}$ (cf Fig. 1a).

2. stay at the current location and capture the first interval in $\delta_i$ and then move to the resting point of sector $\delta_o \in \hat{\mathbf{O}}$ and capture the third interval (cf. Fig. 1b).

3. move to the resting point of $\delta_o \in \hat{\mathbf{O}}$ and capture both (second and third) intervals of $\delta_o$ (cf. Fig. 1c).

4. move to the resting point of sector $\delta_o \in \hat{\mathbf{O}}$ and capture the second interval of $\delta_o$ and then move to the resting point of another sector, $\delta_p \in \hat{\mathbf{O}}$, to capture its second interval (cf. Fig. 1d). Here, sector $\delta_p$ denotes the sector chosen after **S1** once the defender finishes capturing an interval from the sector $\delta_o$.

*Proof of Lemma 47.* The aim of this proof is to establish that for each of the four types of captured intervals identified above, every defender $V_m \in \mathbf{V}$ is charged at most $3(N-M)$ times. Without loss of generality, suppose that $V_m$ is located at the resting point of sector $\delta_i$ at the $j^{th}$ interval. Let $\mathbf{L}'$ denote the set of sectors that do not contain a defender, i.e., the complement of the set $\mathbf{L}$. Note that $|\mathbf{L}'| = N - M$.

We first consider type (1) captured intervals $S_i^{j+1}$ and $S_i^{j+2}$. Since defender $V_m$ decides at time instant $j\tilde{D}$ to capture both $S_i^{j+1}$ and $S_i^{j+2}$, it loses $S_l^{j+2}$ and $S_l^{j+3}$, respectively, from the other remaining sectors which do not contain a defender, i.e., $\delta_l \in \mathbf{L}'$. Thus, the two captured intervals are charged $N-M$ times. The next $N-M$ charge is as follows. Since defender $V_m$ is currently located at $(x_i, \alpha_i)$, it follows that the comparison **S1** must have yielded sector $\delta_i$ at time instant $(j-2)\tilde{D}$. Recall that **S1** requires at least $S_i^j$ and $S_i^{j+1}$ for comparison. Since defender $V_m$ captured $S_i^{j+1}$, the captured intervals are charged $N-M$ times for all $S_l^{j+1}$ intervals, where $\delta_l \in L'$. Thus, the total charge is $2(N-M)$.

Analogously in type (2), captured interval $S_i^{j+1}$ is charged $N-M$ times for lost intervals $S_l^{j+1}$ for all sectors $\delta_l \in \mathbf{L}'$. Two cases arise for the remaining charge:

**Case 1**: The captured intervals are $S_i^{j+1}$ and $S_o^{j+3}$. Then, one charge for lost intervals $S_o^{j+2}, S_i^{j+2}$ and $S_i^{j+3}$ combined as the defender captured $S_i^{j+1}$ instead of $S_o^{j+2}$ and $N-M-1$ charge for all $S_l^{j+2}$ and $S_l^{j+3}$ combined for all sectors $\delta_l \in \mathbf{L}' \setminus \{o\}$. Further, the captured intervals are also charged $N-M$ times for the lost intervals $S_l^{j+4}$ for all sectors $\delta_l \in \mathbf{L}'$.

**Case 2**: The captured intervals are $S_i^{j+1}$ and $S_p^{j+5}$. This case arises when the defender moves to sector $\delta_o$, after capturing $S_i^{j+1}$ (point (ii-b) of **S2**) and then moves to another sector $\delta_p$ and captures $S_p^{j+5}$ (point (ii-a) of **S2**). Without loss of generality, we assume that $p = i$. In this case, one charge for lost intervals $S_o^{j+2}, S_i^{j+2}$ and $S_i^{j+3}$ combined as the defender captured $S_i^{j+1}$ instead of $S_o^{j+2}$ and $N-M-1$ charge for all $S_l^{j+2}$ and $S_l^{j+3}$ combined for all sectors $\delta_l \in \mathbf{L}' \setminus \{o\}$. Finally, one charge is for lost intervals $S_o^{j+3}, S_o^{j+4}$ and $S_o^{j+5}$ combined and $N-M-1$ charge for the lost intervals $S_l^{j+4}$ and $S_l^{j+5}$ combined for sectors $\delta_l \in \mathbf{L}'$. Thus, the total charge for type (2) captured intervals is $3(N-M)$.

We now consider type (3) captured intervals. Type (3) captured intervals $S_o^{j+2}$ and $S_o^{j+3}$ are charged once for the lost intervals $S_i^{j+1}, S_i^{j+2}$ and $S_i^{j+3}$ combined and $N-M-1$ times for the lost intervals $S_l^{j+2}$ and $S_l^{j+3}$ of all sectors $\delta_l \in \mathbf{L}'$. The captured intervals are charged another $N-M$ times for the lost intervals $S_l^{j+4}$ of sectors $\delta_l \in \mathbf{L}$. Finally, the last $N-M$ charge is for the lost intervals $S_l^j$ and $S_l^{j+1}$ as defender $V_m$ captured $S_o^{j+2}$ and not $S_i^{j+1}$.

Finally, we consider type (4) captured intervals $S_o^{j+2}$ and $S_p^{j+4}$. For ease of presentation and without loss of generality, we assume $p = i$. In other words, the defender moves from sector $\delta_i$ to sector $\delta_o$ to capture $S_o^{j+2}$. Then, defender moves back to sector $\delta_i$ to capture $S_i^{j+4}$. Type (4) captured intervals are charged once for the lost intervals $S_i^{j+1}, S_i^{j+2}$ and $S_i^{j+3}$ combined and $N-M-1$ times for the lost intervals $S_l^{j+2}$ and $S_l^{j+3}$ of sectors $\delta_l \in \mathbf{L}'$. Similarly, captured intervals are charged once for the lost intervals $S_o^{j+3}, S_o^{j+4}$ and $S_o^{j+5}$ combined and $N-1-M$ times for the lost intervals $S_l^{j+4}$ and $S_l^{j+5}$ of sectors $\delta_l \in \mathbf{L}'$. Finally, the last $N-M$ charge is for the lost intervals $S_l^j$ and $S_l^{j+1}$ combined as defender $V_m$ captured $S_o^{j+2}$ and not $S_i^{j+1}$.

Since we have shown that each type of captured intervals are charged $3(N-M)$ times, the result is established. □

*Proof of Lemma 48.* Since Coop-SNP directs every defender $V_m$ to stay at a resting point of a sector for some time interval, it can be viewed as a sequence of *traces*, in which each defender $V_m$ spends some number of intervals at one resting point and some number of intervals at another. Observe that a trace of Coop-SNP is composed of $M$ *vehicular traces*, i.e., a trace for an individual defender. A vehicular trace for defender $V_m$ is defined by a set $\{k_1, k_2, \ldots, k_N\}$, where each element $k_l$, $l \in \{1, \ldots, N\}$ denotes the number of intervals that the defender $V_m$ decides to capture by staying at the corresponding resting point of the sector $\delta_l$.

At any time instant $2j\tilde{D}, j > 0$, consider that $V_m$ is located at the resting point of sector $\delta_i$. Observe that any realization of a vehicular trace for defender $v_m$ can be achieved by the combination of one or more *atomic traces* as described in the following cases. Case (i) $k_i = 3$

and $k_l = 0\ \forall l \in \{1, \ldots, N\}$ which satisfy $\delta_l \in \mathbf{L}'$, Case (ii) $k_i = 3$ and $k_o = 1$, Case (iii) $k_i = 2$, $k_o = 1$, case (iv) $k_i = 1$ and $k_o = 1$, and Case (v) $k_i = 0$ and $k_o = 1$. Here, $\delta_o \in \hat{\mathbf{O}}$ denotes the sector that was selected after **S1** and assigned to defender $V_m$ after **S2**.

**Case (i)**: $k_i = 3$ and $k_l = 0$. This case means that the defender stays at its current location for three consecutive intervals and does not change its sector after capturing the three intervals. The only way defender $V_m$ does so is if $\delta_i \in \tilde{\mathbf{O}}$. This implies that, at time instant $2j\tilde{D}$ as well as at $2(j+1)\tilde{D}$, the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds for all $p$ such that $\delta_p \in \mathbf{O}'$. Since the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $2j\tilde{D}$ and given that the defender captures all three intervals $S_i^{j+1}, S_i^{j+2}$ and $S_i^{j+3}$, the lost intervals $S_p^{j+2}$ and $S_p^{j+3}$ for all sectors $\delta_p \in \mathbf{O}'$ are accounted for. Further, as explained in Lemma 46, lost intervals $S_p^{j+1}$ are all accounted as the defender captured $S_i^{j+1}$.

**Case (ii):** $k_i = 3$ and $k_o = 1$. This means that at time $2j\tilde{D}$, the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds (point (i) of **S2**), for all $p$ such that $\delta_p \in \mathbf{O}'$ and, since the defender moved to sector $\delta_o \in \hat{\mathbf{O}}$ after capturing the third interval in sector $\delta_i$, it follows that at time instant $2(j+1)\tilde{D}$, the conditions $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$, for all $p$ such that $\delta_p \in \mathbf{O}'$, and $S_o^{j+4} < S_i^{j+3}$ hold (point (ii-b) of **S2**). As defender captured $S_i^{j+1}$, the lost intervals $S_p^{j+1}$ of sectors $\delta_p \in \mathbf{O}'$, are accounted for. Second, given that the condition $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $(j+2)\tilde{D}$, if the defender captured $S_o^{j+4}$ and $S_o^{j+5}$, then the lost intervals $S_p^{j+4}$, $S_p^{j+5}$, $S_i^{j+3}, S_i^{j+4}$ and $S_i^{j+5}$ would have been accounted for. However, note that the defender captured $S_i^{j+3}$ and $S_o^{j+5}$. Since $S_o^{j+4} < S_i^{j+3}$, the captured intervals account for all these lost intervals. Finally, since the defender captured $S_i^{j+3}, S_i^{j+4}$ and $S_i^{j+5}$, the lost intervals $S_p^{j+2}$ and $S_p^{j+3}$ are accounted as the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $2j\tilde{D}$.

**Case (iii):** $k_i = 2$, $k_o = 1$. This means that at time instant $2j\tilde{D}$, the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds (point (i) of **S2**), for all $p$ such that $\delta_p \in \mathbf{O}'$, and at time instant $2(j+1)\tilde{D}$, $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ hold (point (ii-a) of **S2**). To account for the lost intervals $S_l^{j+2}$ and $S_l^{j+3}$, the defender was supposed to capture $S_i^{j+1}, S_i^{j+2}$ and $S_i^{j+3}$. While the defender captured $S_i^{j+1}$ and $S_i^{j+2}$, it did not capture $S_i^{j+3}$. Given that $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ and $S_o^{j+4} \geq S_i^{j+3}$ holds at time $2(j+1)\tilde{D}$, capturing

$S_i^{j+1}, S_i^{j+2}$ and $S_o^{j+4}$ accounts for all the lost intervals $S_i^{j+1}$, $S_l^{j+2}$ and $S_l^{j+3}$. Now, if the defender, at time instant $2(j+2)\tilde{D}$, decides to stay and capture $S_o^{j+5}$, then lost intervals $S_p^{j+4}$ and $S_p^{j+5}$ are accounted for by either case (i) or case (ii). Otherwise, they are accounted for by case (iv) which we consider next.

**Case (iv):** $k_i = 1$ and $k_o = 1$. There are two sub-cases that arise.

**S**ub-case (a): This sub-case corresponds to when the defender moved from a sector, say $\delta_q$, to sector $\delta_i$ and captured an interval after arrival (point (ii-a) of **S2**). Then, the defender moved to sector $\delta_o$ and captured an interval (point (ii-a) of **S2**). This means that at time instant $2j\tilde{D}$, **S2** assigned sector $\delta_i$, i.e., condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds, for all $p$ such that $\delta_p \in \mathbf{O}'$. To account for lost intervals $S_q^{j+1}, S_q^{j+2}$, $S_q^{j+3}$, $S_p^{j+2}$ and $S_p^{j+3}$, the defender was supposed to capture $S_i^{j+2}$ and $S_i^{j+3}$. Note that the defender captured $S_i^{j+2}$ but not $S_i^{j+3}$ as $k_i = 1$. Since defender moved to sector $\delta_o$, it follows that at time $2(j+1)\tilde{D}$, the condition $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ and $S_o^{j+4} \geq S_i^{j+3}$ holds. As defender captured $S_o^{j+4}$ instead of $S_i^{j+3}$, the lost intervals $S_q^{j+1}, S_q^{j+2}$, $S_q^{j+3}$, $S_p^{j+2}$ and $S_p^{j+3}$ are accounted for. Further, as the defender captured $S_i^{j+2}$ instead of $S_q^{j+1}$, all lost intervals $S_p^{j+1}$ are accounted.

**S**ub-case (b): At time instant $2j\tilde{D}$, the condition $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$, for all $p$ such that $\delta_p \in \mathbf{O}'$, and $S_i^{j+1} > S_o^{j+2}$ holds (point (ii-b) of **S2**). In this case, the defender first captures $S_i^{j+1}$ and then moves to sector $\delta_o$. As $k_o = 1$, it follows that the condition $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $2(j+1)\tilde{D}$. As condition $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $2j\tilde{D}$, capturing $S_o^{j+2}$ and $S_o^{j+3}$ would have accounted for lost intervals $S_i^{j+1}, S_i^{j+2}, S_i^{j+3}, S_p^{j+2}$ and $S_p^{j+3}$. Although the defender captured $S_o^{j+3}$, it did not capture $S_o^{j+2}$. Given that $S_i^{j+1} > S_o^{j+2}$ holds and that the defender captured $S_i^{j+1}$, the lost intervals $S_o^{j+2}, S_i^{j+2}, S_i^{j+3}, S_p^{j+2}$ and $S_p^{j+3}$ are accounted for. Further, capturing $S_i^{j+1}$ also accounts for the lost intervals $S_p^{j+1}$.

**Case (v):** $k_i = 0$ and $k_o = 1$. At time instant $2(j-1)\tilde{D}$ let the defender be located in sector $\delta_q$. Then, this case is only possible if, at time instant $2(j-1)\tilde{D}$, the conditions $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$, for all $p$ such that $\delta_p \in \mathbf{O}'$, and $S_q^{j-1} > S_i^j$ both hold (point (ii-b) of **S2**) and at time instant $2j\tilde{D}$, the conditions $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ and $S_o^{j+2} \geq S_i^{j+1}$ hold (point (ii-a) of **S2**), for all $p$

such that $\delta_p \in \mathbf{O}'$. Thus, the defender does not capture any interval in sector $\delta_i$. We account for the lost intervals as follows.

Given that the condition $\eta^i_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ holds at time instant $2(j-1)\tilde{D}$, capturing $S_i^j$ and $S_i^{j+1}$ would have accounted for intervals $S_q^{j-1}, S_q^j, S_q^{j+1}, S_p^j$ and $S_p^{j+1}$. Further, since conditions $\eta^o_{\mathbf{L}} \geq \eta^p_{\mathbf{L}}$ and $S_o^{j+2} \geq S_i^{j+1}$ both hold at time instant $2j\tilde{D}$, capturing $S_o^{j+2}$ and $S_o^{j+3}$ would account for $S_i^{j+1}, S_i^{j+2}, S_i^{j+3}, S_p^{j+2}$ and $S_p^{j+3}$. Given that conditions $S_q^{j-1} \geq S_i^j$ and $S_o^{j+2} \geq S_i^{j+1}$ hold and the defender captures $S_q^{j-1}$ and $S_o^{j+2}$, the lost intervals $S_i^j, S_q^j, S_q^{j+1}, S_p^j$ and $S_p^{j+1}$ are accounted. Further, lost intervals $S_i^{j+1}, S_i^{j+2}, S_i^{j+3}, S_p^{j+2}$ and $S_p^{j+3}$ are accounted based on whether the defender decides to capture $S_o^{j+3}$ (case (i), case (ii) or case (iii)) or if it moves to some other sector (case (iv)).

Finally, note that the boundary cases of the first and the last intervals fall into these cases by adding dummy intervals of zero cardinality $S_i^0, \forall i \in \{1, \ldots, N\}$ and $S_i^{Y+1}$, where $Y$ denotes the last interval that consists of intruders in any sector. We assume that the defender captures all of the dummy intervals. This concludes the proof. □

## APPENDIX B: EXPRESSIONS FOR MATRICES

We now provide the expression for the matrices $P, Q_j, M$ and $L$, respectively. For ease of notation, denote $a_i = \hat{X}_i^t$, $b_i = \hat{Y}_i^t$. Further, let $\mathbf{I}_{n\times p}$ (resp. $\mathbf{0}_{n\times p}$) denote the identity (resp. zero) matrix of dimension $n \times p$. Then,

$$P = \frac{1}{\sigma_V^2} \times \begin{bmatrix} \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 8} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{1\times 6} & \delta\sigma_V^2 & -\delta\sigma_V^2 & \mathbf{0}_{1\times 8} & 0 \\ \mathbf{0}_{1\times 6} & -\delta\sigma_V^2 & \delta\sigma_V^2 & \mathbf{0}_{1\times 8} & 0 \\ \mathbf{0}_{8\times 6} & \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 1} \\ \mathbf{0}_{1\times 6} & 0 & 0 & \mathbf{0}_{1\times 8} & 1 \end{bmatrix}, F = \begin{bmatrix} F_1 & \mathbf{0}_{8\times 9} \\ \mathbf{0}_{9\times 8} & F_2 \end{bmatrix},$$

where $F_1 =$

$$\begin{bmatrix} b_2^2 & -a_2b_2 & -b_1b_2 & 2a_1b_2 - a_2b_1 & -b_2 & b_2 & 0 & a_2b_1b_2 - a_1b_2^2 \\ -a_2b_2 & a_2^2 & 2a_2b_1 - a_1b_2 & -a_1a_2 & a_2 & -a_2 & 0 & a_1a_2b_2 - b_1a_2^2 \\ -b_1b_2 & 2a_2b_1 - a_1b_2 & b_1^2 & -a_1b_1 & b_1 & -b_1 & 0 & a_1b_1b_2 - a_2b_1^2 \\ 2a_1b_2 - a_2b_1 & -a_1a_2 & -a_1b_1 & a_1^2 & -a_1 & a_1 & 0 & a_1a_2b_1 - b_2a_1^2 \\ -b_2 & a_2 & b_1 & -a_1 & 1 & -1 & 0 & 0 \\ b_2 & -a_2 & -b_1 & a_1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2b_1b_2 - a_1b_2^2 & a_1a_2b_2 - b_1a_2^2 & a_1b_1b_2 - a_2b_1^2 & a_1a_2b_1 - b_2a_1^2 & 0 & 0 & 0 & (a_1b_2 - a_2b_1)^2 \end{bmatrix}$$

and $F_2 =$

$$\begin{bmatrix} -(a_2^2+b_2^2) & 0 & -2a_1a_2 & -2a_1b_2 & b_2 & 0 & a_2 & 0 & a_1(a_2^2+b_2^2) \\ 0 & -(a_2^2+b_2^2) & -2a_2b_1 & -2b_1b_2 & 0 & a_2 & 0 & b_2 & b_1(a_2^2+b_2^2) \\ -2a_1a_2 & -2a_2b_1 & -(a_1^2+b_1^2) & 0 & 0 & b_1 & a_1 & 0 & a_2(a_1^2+b_1^2) \\ -2a_1b_2 & -2b_1b_2 & 0 & -(a_1^2+b_1^2) & a_1 & 0 & 0 & b_1 & b_2(a_1^2+b_1^2) \\ b_2 & 0 & 0 & a_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_1 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_2 & 0 & a_1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & b_2 & 0 & b_1 & 0 & 0 & 0 & 1 & 0 \\ a_1(a_2^2+b_2^2) & b_1(a_2^2+b_2^2) & a_2(a_1^2+b_1^2) & b_2(a_1^2+b_1^2) & 0 & 0 & 0 & 0 & -(a_1^2+b_1^2)(a_2^2+b_2^2) \end{bmatrix}.$$

Moreover,

$$Q_1 = \begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{0}_{2\times6} & \mathbf{0}_{2\times9} \\ \mathbf{0}_{5\times4} & \mathbf{0}_{5\times4} & \mathbf{0}_{5\times9} \\ \mathbf{0}_{1\times7} & -\mu_1^2 & \mathbf{0}_{1\times9} \\ \mathbf{0}_{9\times4} & \mathbf{0}_{9\times4} & \mathbf{0}_{9\times9} \end{bmatrix}, Q_2 = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times9} \\ \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times9} \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{1\times2} & \mathbf{0}_{1\times5} & -\mu_2^2 & \mathbf{0}_{1\times9} \\ \mathbf{0}_{9\times2} & \mathbf{0}_{9\times5} & \mathbf{0}_{9\times1} & \mathbf{0}_{9\times9} \end{bmatrix},$$

$$M_1 = \begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{0}_{2\times5} & [e_1^t - p^t] & \mathbf{0}_{2\times9} \\ \mathbf{0}_{4\times2} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times2} & \mathbf{0}_{4\times9} \\ \mathbf{0}_{1\times6} & -1 & 0 & \mathbf{0}_{1\times9} \\ [e_1^t - p^t]' & \mathbf{0}_{1\times5} & \|e_1^t - p^t\|^2 & \mathbf{0}_{1\times9} \\ \mathbf{0}_{9\times2} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times11} \end{bmatrix},$$

$$M_0 = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times9} \\ \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times3} & [e_2^t - p^t] & \mathbf{0}_{2\times9} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times9} \\ \mathbf{0}_{1\times2} & \mathbf{0}_{1\times4} & -1 & 0 & \mathbf{0}_{1\times9} \\ \mathbf{0}_{1\times2} & [e_2^t - p^t]' & \mathbf{0}_{1\times3} & \|e_2^t - p^t\|^2 & \mathbf{0}_{1\times9} \\ \mathbf{0}_{9\times2} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times9} \end{bmatrix}.$$

We now define the matrices $L_g \in \mathbb{R}^{17\times 17}, \forall g \in \{1,\ldots,10\}$. Let $L_g(k,l)$ denote an element at the $k^{th}$ row and the $l^{th}$ column of the matrix $L_g, g \in \{1,\ldots,10\}$. Then,

$$L_1(k,l) = \begin{cases} 0.5, \text{ if } k=1, l=4, \\ 0.5, \text{ if } k=4, l=1, \\ -0.5, \text{ if } k=5, l=8, \\ -0.5, \text{ if } k=8, l=5, \\ 0 \text{ otherwise} \end{cases}, L_2(k,l) = \begin{cases} 0.5, \text{ if } k=2, l=3, \\ 0.5, \text{ if } k=3, l=2, \\ -0.5, \text{ if } k=6, l=8, \\ -0.5, \text{ if } k=8, l=6, \\ 0 \text{ otherwise} \end{cases},$$

$$L_3(k,l) = \begin{cases} 0.5, \text{ if } k=1, l=17, \\ 0.5, \text{ if } k=17, l=1, \\ -0.5, \text{ if } k=9, l=8, \\ -0.5, \text{ if } k=8, l=9, \\ 0 \text{ otherwise} \end{cases}, L_4(k,l) = \begin{cases} 0.5, \text{ if } k=2, l=17, \\ 0.5, \text{ if } k=17, l=2, \\ -0.5, \text{ if } k=10, l=8, \\ -0.5, \text{ if } k=8, l=10, \\ 0 \text{ otherwise} \end{cases},$$

$$L_5(k,l) = \begin{cases} 0.5, \text{ if } k=3, l=17, \\ 0.5, \text{ if } k=17, l=3, \\ -0.5, \text{ if } k=11, l=8, \\ -0.5, \text{ if } k=8, l=11, \\ 0 \text{ otherwise} \end{cases}, L_6(k,l) = \begin{cases} 0.5, \text{ if } k=4, l=17, \\ 0.5, \text{ if } k=17, l=4, \\ -0.5, \text{ if } k=12, l=8, \\ -0.5, \text{ if } k=8, l=12, \\ 0 \text{ otherwise} \end{cases},$$

$$L_7(k,l) = \begin{cases} 0.5, \text{ if } k = 5, l = 17, \\ 0.5, \text{ if } k = 17, l = 5, \\ -0.5, \text{ if } k = 13, l = 8, \\ -0.5, \text{ if } k = 8, l = 13, \\ 0 \text{ otherwise} \end{cases}, L_8(k,l) = \begin{cases} 0.5, \text{ if } k = 6, l = 17, \\ 0.5, \text{ if } k = 17, l = 6, \\ -0.5, \text{ if } k = 14, l = 8, \\ -0.5, \text{ if } k = 8, l = 14, \\ 0 \text{ otherwise} \end{cases},$$

$$L_9(k,l) = \begin{cases} 0.5, \text{ if } k = 3, l = 9, \\ 0.5, \text{ if } k = 9, l = 3, \\ -0.5, \text{ if } k = 15, l = 8, \\ -0.5, \text{ if } k = 8, l = 15, \\ 0 \text{ otherwise} \end{cases}, L_{10}(k,l) = \begin{cases} 0.5, \text{ if } k = 4, l = 10, \\ 0.5, \text{ if } k = 10, l = 4, \\ -0.5, \text{ if } k = 16, l = 8, \\ -0.5, \text{ if } k = 8, l = 16, \\ 0 \text{ otherwise} \end{cases},$$