Competitive Perimeter Defense of Conical Environments

Shivam Bajaj\textsuperscript{1}, Eric Tornq\textsuperscript{2}, Shaunak D. Bopardikar\textsuperscript{1}, Alexander Von Moll\textsuperscript{3}, Isaac Weintraub\textsuperscript{3}, Eloy Garcia\textsuperscript{3}, David W. Casbeer\textsuperscript{3}

Abstract—We consider a perimeter defense problem in a planar conical environment in which a single vehicle, having a finite capture radius, aims to defend a concentric perimeter from mobile intruders. The intruders are arbitrarily released at the circumference of the environment and they move radially toward the perimeter with fixed speed. We present a competitive analysis approach to this problem by measuring the performance of multiple online algorithms for the vehicle against arbitrary inputs, relative to an optimal offline algorithm that has information about entire input instance in advance. In particular, we establish two necessary conditions on the parameter space to guarantee (i) finite competitiveness of any algorithm and (ii) a competitive ratio of at least 2 for any algorithm. We then design and analyze three online algorithms and characterize parameter regimes in which they have finite competitive ratios. Specifically, our first two algorithms are provably 1, and 2-competitive, respectively, whereas our third algorithm exhibits different competitive ratios in different regimes of problem parameters. Finally, we provide a numerical plot in the parameter space to reveal additional insights into the relative performance of our algorithms.

I. INTRODUCTION

This work considers a perimeter defense problem in a conical environment involving a single vehicle that seeks to intercept mobile intruders before they enter a specified region (referred to as the perimeter). This scenario arises when a UAV is required to tag (or relay critical information to) intruders (targets) before they reach a specific region of interest. The intruders are generated at the boundary of the environment and move radially inwards with fixed speed toward the perimeter. The vehicle, which has a finite capture radius, moves with bounded speed (greater than that of the intruders) with the aim of capturing as many intruders as possible before they reach the perimeter. This is an online problem as the number and the arrival location of intruders is sequentially revealed over time.

Prior works in the area of perimeter defense have either focused on determining optimal strategies of small number of agents or consider a stochastic arrival process for the intruders [1]–[3]. Although these studies provide valuable insights, they do not address the worst-case performance where the intruders might coordinate their actions [4]. In this work, we adopt a competitive analysis technique [5], to assess online vehicle motion planning algorithms in the worst-case. In competitive analysis, we measure the performance of an online algorithm, using the concept of competitive ratio, which we formally define in Section II.

A related area of research is vehicle routing in which inputs become available over time. Introduced on graphs in [6], a typical approach requires that the vehicle routes be re-planned as new information is revealed over time. We refer the reader to [7] and the references therein for a review of this literature. In most of the vehicle routing problems, the input (known as demands) are static, and so, the problem is to find the shortest route through the demands in order to minimize (maximize) the cost (reward) such as total time or number of inputs serviced. However, in perimeter defense scenarios, the input (intruders) are not static. Instead, they are moving towards a specified region and thus, this problem is more challenging than the former. In our previous works, we introduced perimeter defense problems in circular and rectangular environments with stochastically generated input, [3], [8]. The key distinction of this work from the past works is the characterization of competitiveness for the worst-case inputs, as opposed to the stochastically generated inputs.

Perimeter defense problems were first introduced for a single vehicle and a single intruder in [9]. Since then, perimeter defense has been mostly formulated as a pursuit-evasion differential game. The multiplayer setting for the same has been studied extensively as a reach-avoid game in which the aim is to design control policies for the intruders and the defenders [10]–[12]. A typical approach requires computing solutions to the Hamilton-Jacobi-Bellman-Isaacs equation, which is generally only suitable for low dimensional state spaces and in simple environments [13], [14]. Recent works include [15]–[18]. Authors in [19] propose a receding horizon strategy based on maximum matching, [16], [17] consider a scenario wherein the defenders are constrained to be on the perimeter and [18] extends the reach avoid game to $n$-dimensional Euclidean spaces. Previously, we introduced a perimeter defense problem for linear environments based on the use of competitive analysis [20]. The key distinction of this work from [20] is the geometry of the environment which yields novel results in terms of optimally placing the vehicle, role of capture radius and additional conditions to ensure competitiveness of the algorithms.

The general contribution of this paper is that we consider a conical environment of unit radius and angle $2\theta$ in which arbitrary number of intruders are released at the circumference of the environment at arbitrary time instances. Upon release, the intruders move radially inwards with fixed speed $v < 1$ with the aim of reaching a conical perimeter of radius
$\rho < 1$ and angle $2\theta$. A single vehicle having a finite capture radius $r$, moves with maximum speed of unity with an aim to capture the intruders. Our main contributions are as follows. We first establish two necessary conditions in the parameter space for achieving a $c$-competitive algorithm with a finite $c$. Specifically, we characterize the parameter regime in which no online algorithm is $c$-competitive and a parameter regime in which no algorithm can be better than 2-competitive. Next, we design and analyze three classes of algorithms and establish their competitiveness. Specifically, we identify parameter regimes in which the first two algorithms are provably 1 and 2-competitive, respectively, and the third algorithm has a finite competitive ratio that varies with the problem parameters ($r, \rho, \theta$).

This paper is organized as follows. In section II, we formally describe our problem and define competitive ratio for online algorithms. Section III establishes the necessary conditions. In section IV, we design and analyze three algorithms and establish their competitive ratios, section V provides additional insights through numerous parameter space plots and finally, section VI summarizes this work and outlines directions for future works. For brevity, we only provide an outline for some of our intermediate results. The detailed proofs of all results are available in [21].

II. PROBLEM DESCRIPTION

Consider a conical environment of $E(\theta) = \{(y, \alpha) : 0 < y \leq 1, -\theta \leq \alpha \leq \theta \}$ which contains a conical region (referred to as perimeter) $R(\rho, \theta) = \{(z, \alpha) : 0 < z \leq \rho, 1 - \theta \leq \alpha \leq \theta \}$, where $\theta$ is measured with respect to $y$-axis. 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 $v$ towards the origin in order to breach the perimeter. The defense consists of a single vehicle with motion modeled as a first order integrator\(^1\) with a maximum speed of unity and a finite capture radius $r < \rho$. A capture circle is defined as a circle of radius $r$, centered at the vehicle’s location. An intruder is captured and subsequently removed from $E(\theta)$ if it lies within or on the capture circle. An intruder is lost if it reaches the perimeter without being captured by the vehicle.

A problem instance $P(\theta, \rho, v, r)$ is characterized by four parameters: the speed of the intruders, $v < 1$, the perimeter’s radius $0 < \rho < 1$, the angle that defines the size of the environment as well as the perimeter, $0 < \theta \leq \pi$ and, the capture radius $r < \rho$. An input instance $I$ is a set of tuples consisting of time instant $t \leq T$, where $T$ denotes the final time instant, the number of intruders $N(t)$ that are released at time instant $t$, and the arrival location of each of the $N(t)$ intruders. Formally, $I = \{t, N(t), \{(1, \alpha_1), (1, \alpha_2), \ldots , (1, \alpha_{N(t)})\}\}_{t=0}^{T}$, for any $\alpha_l \in [-\theta, 0]$, where $1 \leq l \leq N(t)$.

An online algorithm $A$ assigns a velocity with at most unit magnitude to the vehicle as a function of the input $I(t) \subset I$ revealed until time $t$, yielding the kinematic model, $\dot{x}(t) = A(I(t))$, where $x$ denotes the vehicle’s polar coordinates. An optimal offline algorithm is a non-causal algorithm which computes the velocity of the vehicle at any time $t$ having the information of the entire input instance $I$.

Definition 1 (Competitive Ratio) Given a problem instance $P(\theta, \rho, v)$, an input instance $I$, and an online algorithm $A$, let $A(I)$ denote the number of intruders captured by the vehicle when using $A$ on input instance $I$. Let $O$ denote the optimal offline algorithm that maximizes the number of intruders captured out of input instance $I$. Then, the competitive ratio of $A$ on $I$ is defined as $c_{A}(I) = O(I)/A(I) \geq 1$, and the competitive ratio of $A$ for the problem instance $P$ is $c_{A}(P) = \sup_{I} c_{A}(I)$. Finally, the competitive ratio for the problem instance $P$ is $c(P) = \inf_{A} c_{A}(P)$. An algorithm is $c$-competitive for the problem instance $P(\theta, \rho, v)$ if $c_{A}(P) \leq c$, where $c \geq 1$ is a constant.

Problem Statement: The aim is to establish fundamental guarantees and to design $c$-competitive algorithms for the vehicle with minimum $c$.

In light of Lemma 1 in [20], it suffices to restrict to extreme speed algorithms that either move the vehicle with maximum speed, i.e., unity, or keep it stationary.

III. FUNDAMENTAL LIMIT FOR FINITE $c$

We will first establish necessary conditions in the space of problem parameters ($\theta, v, r, \rho$) for finite $c$. We begin by providing two properties based on geometry of the environment. The proof follows directly from the geometry and has been omitted for brevity (cf. [21] for a complete proof).

Lemma III.1 For a problem instance $P(\theta, \rho, v)$ with $\theta < \pi/2$, all intruders can be captured if $r \geq \rho \tan(\theta)$ by positioning the vehicle at $\left(\frac{\rho}{\cos(\theta)}, 0\right)$.

We now characterize the minimum time required by the vehicle to move from one end of the perimeter to the other.

Lemma III.2 The minimum time required by the vehicle 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 $\theta < \pi/2$ and $2(\rho - r)$, otherwise.

We now present our first necessary condition on the problem parameters for a finite $c(P)$.

Theorem III.3 (Necessary condition for finite $c(P)$) For any problem instance $P(\theta, r, v, \rho)$ with parameters satisfying

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

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

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

Proof: In this proof, we first construct an input instance and then determine the number of intruders captured in that

---

\(^1\)The techniques and analysis used in this work can be extended to other models such as double integrator, and would be addressed in a future work.
input instance by any online algorithm $A$ as well as the optimal offline algorithm $O$.

For both online and optimal offline algorithms, assume that the vehicle starts at the origin at time 0. The input instance 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 $A$ never captures any stream intruders, the stream never ends meaning the algorithm $A$ will not be $c$-competitive for any constant $c \geq 1$, and the first result follows as the optimal offline algorithm can move to $(\rho, \theta)$ and capture all the stream intruders. We thus assume $A$ does capture at least one stream intruder, say the $i$th one, at time $t_i$. 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 $t_i$.

We now identify how many intruders $A$ 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 requires the perimeter and thus is lost just as the next stream intruder arrives. We now show that the vehicle cannot capture any of the $c + 1$ burst intruders. At time $t_i$, the vehicle must be at most $r$ distance away from the $i$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 $t_i$. From Lemma III.2 and our given conditions, $2(\rho \sin(\theta) - r) > \frac{1 - \rho}{v}$ (resp. $2(\rho \sin(\theta) - r) > \frac{1 - \rho}{v}$) for $\theta < \frac{\pi}{2}$ (resp. $\theta \geq \frac{\pi}{2}$), the vehicle is ensured to lose the burst intruders.

On the other hand, the optimal offline algorithm $O$ can move the vehicle to location $(x, \alpha)$, as defined in Lemma III.2, until the first $i - 1$ intruders have been captured and then move the vehicle to $(x, -\alpha)$ capturing the burst intruders, losing only the $i$th intruder. This concludes the proof.

We now establish a necessary condition for the existence of online algorithms having a competitive ratio of at least 2. We first characterize locations $(t_1, \alpha_1) \in \mathcal{E}(\theta)$ and $(t_2, \alpha_2) \in \mathcal{E}(\theta)$ for the vehicle (Fig. 1), where

$$t_1 = \sqrt{\frac{1 + r^2 - 2r(1 - \rho \cos(2\theta))}{\sqrt{1 + r^2 - 2r \cos(2\theta)}}},$$

$$\alpha_1 = \tan^{-1} \left( \frac{\sin(\theta) \sqrt{1 + r^2 - 2r \cos(2\theta)} - r(1 + \rho) \sin(\theta)}{\cos(\theta) \sqrt{1 + r^2 - 2r \cos(2\theta)} - r(1 - \rho) \cos(\theta)} \right),$$

$$t_2 = \sqrt{\frac{r^2 + r^2 + 2r \rho \cos(2\theta) - \rho}{\sqrt{1 + r^2 - 2r \cos(2\theta)}}},$$

$$\alpha_2 = \tan^{-1} \left( \frac{-\rho \sin(\theta) \sqrt{1 + r^2 - 2r \cos(2\theta)} + r(1 + \rho) \sin(\theta)}{\rho \cos(\theta) \sqrt{1 + r^2 - 2r \cos(2\theta)} + r(1 - \rho) \cos(\theta)} \right).$$

These locations are determined analogously to the proof of Lemma III.2 and is omitted for brevity (see [21]).

**Theorem III.4 (Necessary condition for $c(P) \geq 2$)** For any problem instance $P(\theta, r, \rho, v)$, $c(P) \geq 2$ if

$$\frac{1 - \rho}{v} \leq \sqrt{1 + \rho^2 - 2\rho \cos(2\theta) - 2r}, \quad \text{if } \theta \leq \frac{\pi}{2},$$

$$\frac{1 - \rho}{v} \leq 1 + \rho - 2r, \quad \text{if } \theta > \frac{\pi}{2}. $$

**Proof:** The key idea is to construct input instances for which any online algorithm is guaranteed to lose half the number of intruders out of that instance, while proving that an offline algorithm exists that can intercept all intruders. All of our input instances consist of two intruders denoted by $a$ and $b$ released at locations $(1, \theta)$ and $(1, -\theta)$, respectively, and we assume that the vehicle starts at the origin. Two cases arise; (i) $\theta \leq \frac{\pi}{2}$, and (ii) $\theta > \frac{\pi}{2}$.

- **Case (i):** Suppose that $\frac{1 - \rho}{v} = \sqrt{1 + \rho^2 - 2\rho \cos(2\theta) - 2r}$.

  Consider an input instance $I_1$ in which both intruders $a$ and $b$ are released at time instant $t_1$. This is the time that the vehicle 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 vehicle to capture both the intruders in the worst case is $\sqrt{1 + \rho^2 - 2\rho \cos(2\theta) - 2r}$, 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 through geometry and is omitted for brevity (refer [21]). As $\frac{1 - \rho}{v} = \sqrt{1 + \rho^2 - 2\rho \cos(2\theta) - 2r}$ is a 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 following 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 instance $I_1$. At time 0, the vehicle 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 vehicle 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 vehicle 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 vehicle (Fig. 1). Thus, any algorithm that hopes to be better than 2-competitive must capture both the intruders in this input instance and the only way to do so is to move to location $(t_1, \alpha_1)$ or $(t_1, -\alpha_1)$ arriving exactly at time $t_1$.

  Now consider input instances $I_2$ and $I_3$. In $I_2$, intruder $a$ arrives at time $t_1$ and intruder $b$ arrives at time $t_1 + \epsilon$, where $\epsilon < L = 2\sin(\theta) \left(1 - \frac{1 - \rho}{v(1 + \rho)}\right)$ and $L$ denotes the minimum time required by the vehicle 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 instance $I_2$ (resp $I_3$) are constructed for algorithms that have the vehicle
arriving at location \((t_1, -\alpha_1)\) (resp. \((t_1, \alpha_1)\)) at time \(t_1\). Any algorithm that has the vehicle arriving at location \((t_1, -\alpha_1)\) (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 instance \(I_3\). This follows as the vehicle can capture intruder \(b\) if it moves directly to location \((t_2, -\alpha_2)\) (Fig. 1a). However, as intruder \(a\) arrives in at most \(\epsilon < L\) time units, the vehicle will not be able to capture intruder \(a\) (Fig. 1b). An optimal offline algorithm can capture both the intruders simply by moving to \((t_1, -\alpha_1)\) at time \(t_1\), capturing intruder \(b\) upon arrival and then to \((t_2, -\alpha_2)\) to capture intruder \(a\).

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

**Case (ii):** \(\theta > \frac{\pi}{2}\). Except for when \(\theta = \pi\), the vehicle must move first to the origin and then to the next intercept point. Note that, the vehicle 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 \((\rho-r, -\theta)\). Following similar steps as case (i), we construct input instances \(I_1, \ldots, I_5\) (omitted for brevity) and show that no online algorithm can capture both the intruders from those input instances.

In summary, even restricting our input instance 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 that provide sufficient conditions on the competitive ratios.

### IV. Algorithms

We start by defining an angular path for the vehicle. Let the vehicle be located at \((x, \alpha) \in \mathcal{E}(\theta)\) for any \(0 < x \leq 1\) and \(\alpha \in [-\theta, \theta]\). An angular path is a circular arc centered at the origin defined as \(T(x, \beta, \bar{\beta}) := \{(x, \beta) : \beta \leq \beta \leq \bar{\beta}\}\) for any \(\beta, \bar{\beta} \in [-\theta, \theta]\) such that \(\beta \leq \alpha \leq \bar{\beta}\) and \(\beta \neq \bar{\beta}\). We say that the vehicle completes its motion on the angular path when the vehicle returns to its starting location after moving along all of the points in \(T\) twice. Once to move from the starting location \((x, \alpha)\) to \((x, \bar{\beta})\) (resp. \((x, \beta)\)), and second, to move from location \((x, \bar{\beta})\) (resp. \((x, \beta)\)) to location \((x, \beta)\) (resp. \((x, \bar{\beta})\)) and then back to the starting location \((x, \alpha)\).

**A. Angular Sweep algorithm**

Angular Sweep is an open loop algorithm, described as follows. The vehicle starts at location \((x_S, 0)\), where \(x_S \in [\frac{1-\theta}{\rho v}, \min\{1 - r, \rho + r\}]\), and \(a = 1\) if \(\theta = \pi\) and \(a = 4\) if \(\theta \neq \pi\). This choice for the location \(x_S\) is justified in [21]. In Angular Sweep, the vehicle moves on an angular path with \(x = x_S, \beta = -\theta\) and \(\bar{\beta} = \theta\) for any \(\theta < \pi\). For \(\theta = \pi\), the vehicle moves on a circle with \(x_S\) as the radius and the origin as the center.

We first define the angular sweep algorithm for \(\theta \neq \pi\). At time \(0\), the vehicle first picks a velocity with unit magnitude and direction tangent to the angular path, oriented to the right until it reaches \((x_S, \theta)\). Once it reaches the endpoint, the vehicle switches direction and moves towards the other endpoint, \((x_S, -\theta)\). From this moment on, the vehicle only switches direction after it reaches an endpoint. In other words, the vehicle moves on the angular path \(T(x_S, -\theta, \theta)\), moving towards \((x_S, \theta)\) at time \(0\).

We now define the algorithm for \(\theta = \pi\). At time \(0\), the vehicle picks a velocity with unit magnitude and direction tangent to the angular path, oriented to the right. From this point on, the vehicle keeps on moving in the same direction for the entire duration, i.e., the vehicle moves on a circle of radius \(x_S\) and center as the origin.

**Theorem IV.1 (Angular Sweep competitiveness)** For any problem instance \(\mathcal{P}(r, \rho, \theta, v)\) such that

\[
v \leq \min\left\{\frac{2r}{(\rho + r)\theta}, 1 - \frac{1}{(1-r)\alpha}\right\},
\]

where \(a = 2\) if \((\theta = \pi)\) or \(a = 4\) if \((\theta \neq \pi)\), with the choice of any \(r \leq \left[\frac{\rho}{\rho - r}, \min\{1 - r, \rho + r\}\right]\).

Angular Sweep is \(1\)-competitive. Otherwise, Angular Sweep is not \(c\)-competitive for any constant \(c\).

**Proof:** First, if equation (1) holds, then the interval \([\frac{1-\theta}{\rho v}, \min\{1 - r, \rho + r\}]\) is non-empty and well defined. Thus, it suffices to show that any \(x_S\) from the said interval guarantees that Angular Sweep captures every intruder.

Without loss of generality, we assume that, in the worst-case, at time instant \(t\), the vehicle has just left the location \((x_S, \theta)\) and intruder \(i\) is located at \((x_S + r, \theta)\). The vehicle takes a total of \(a\theta x_S\) time units to return to the location \((x_S, \theta)\) whereas the intruder takes \(\frac{x_S + r - x}{\rho}\) time units to reach the perimeter. Thus, in order to ensure that the intruder \(i\) is captured and takes time no less than \(\frac{x_S + r - x}{\rho}\), we require \(a\theta x_S \leq (x_S + r - x)/\rho\) and \(x_S \leq 1 - r\), respectively, which holds given that \(x_S \in [\frac{1-\theta}{\rho v}, \min\{1 - r, \rho + r\}]\).

For any \(x_S \notin [\frac{1-\theta}{\rho v}, \min\{1 - r, \rho + r\}]\), we can construct an input instance with stream of intruders always arriving at \((1, \theta)\) such that when the vehicle leaves location \((x_S, \theta)\), an intruder is located at \((x_S + r, \theta)\). Since \(x_S \notin [\frac{1-\theta}{\rho v}, \min\{1 - r, \rho + r\}]\), all intruders will be lost and the result follows.

### B. Conical Compare and Capture

We now describe Conical Compare and Capture (ConCaC) algorithm and establish that ConCaC is \(2\)-competitive for parameter regimes beyond those required for Angular Sweep.

An epoch \(k\) is defined as the time interval in which the vehicle completes its motion on angular path with a specified distance \(x_c \in [\frac{1-\theta}{\rho v}, \min\{\rho + r, \frac{1-\theta}{\rho v}\}]\) which is fixed for all epochs. The choice of \(x_c\) is justified in [21]. ConCaC sets the parameters \(\beta\) and \(\bar{\beta}\) for the angular path at the start of every epoch. Denote \(S_{left}^k(v, \rho)\) (resp. \(S_{right}^k(v, \rho)\)) as the total number of intruders in the set \(S_{left}^k\) (resp. \(S_{right}^k\)) in epoch \(k\), where

\[
S_{left}^k(v, \rho) := \left\{(y, \beta) : \beta + \beta x_C v < y \leq \min\{1, x_c + r + (2\theta - \beta) v x_C\} \forall \beta \in [0, \theta]\right\}
\]

and

\[
S_{right}^k(v, \rho) := \left\{(y, \beta) : \beta - \beta x_C v < y \leq \min\{1, x_c + r + (2\theta + \beta) v x_C\} \forall \beta \in (0, -\theta]\right\}.
\]
ConCaC algorithm is defined in Algorithm 1 and is summarized as follows. At the start of every epoch \( k \), the vehicle compares the total number of intruders in the set \( S_{\text{left}}^k \) and \( S_{\text{right}}^k \) (Fig. 2). If \(|S_{\text{left}}^k| < |S_{\text{right}}^k|\) (Line 4 in Algorithm 1), then the vehicle moves on the angular path \( T(x_C, 0, \theta) \) until it reaches location \((x_C, \theta)\) and then returns to \((x_C, 0)\), moving on the same angular path. Otherwise (Line 8), the vehicle moves on an angular path \( T(x_C, -\theta, 0) \) towards the location \((x_C, -\theta)\) and then returns to \((x_C, 0)\) moving on the same angular path. The vehicle then repeats the same for the next epoch.

For the initial case, we assume time 0 as the time when the first intruder arrives in the environment. The vehicle starts at location \((x_C, 0)\) and waits for \( 1 - \min\{1, x_C + r + 2\theta v_{xC}\} \) amount of time and then begins its first epoch.

**Lemma IV.2** Any intruder that lies beyond\(^2\) the location \((x_C + r + (2\theta - \beta) v_{xC}, \beta), \forall \beta \in [-\theta, \theta]\) in epoch \( k \), will either be contained in the set \( S_{\text{left}}^{k+1} \) or in \( S_{\text{right}}^{k+1} \) in epoch \( k+1 \) and is not lost at the start of epoch \( k+1 \) if \( v \leq \frac{2\theta + \beta}{2\theta v_{xC}} \).

**Proof:** [Sketch] Note that the region beyond \((x_C + r + (2\theta - \beta) v_{xC}, \beta), \forall \beta \in [-\theta, \theta]\) is the region between the circumference of the environment and the sets \( S_{\text{left}}^k \) and \( S_{\text{right}}^k \) in any epoch \( k \) (region between the black solid line and blue dashed curves in Fig. 2). To establish this result, we determine the total time taken by the vehicle in an epoch and the time taken by any intruder that was not contained in the set \( S_{\text{left}}^k \) and \( S_{\text{right}}^k \) in the worst case. The proof then follows from the fact that the time taken by the vehicle must be less than the time taken by the intruder considered.

**Theorem IV.3 (ConCaC competitiveness)** For any problem instance \( \mathcal{P}(\theta, r, \rho, v) \) such that

\[
v \leq \min\left\{ \frac{r}{\theta r + r}, \frac{1 - \rho}{\rho (\theta r + r)} \right\},
\]

with the choice of any \( x_C \in \left[\frac{r - \rho r}{2\theta v_{xC}}, \min\{r + \rho, \frac{1 - \rho r}{1 + r} \}\right] \), ConCaC algorithm is 2-competitive.

**Proof:** First, if equation (2) holds, then the interval \( \left[\frac{r - \rho r}{2\theta v_{xC}}, \min\{r + \rho, \frac{1 - \rho r}{1 + r} \}\right] \) is non-empty. Therefore, it suffices to show that for any \( x_C \) from the said interval, ConCaC algorithm is 2-competitive. Lemma IV.2 ensures that every intruder will belong to either set \( S_{\text{left}}^k \) or \( S_{\text{right}}^k \) in every epoch \( k \). In every epoch \( k \), the vehicle compares the total number of intruders with radial coordinate more than \( x_C + r + (2\theta - \beta) v_{xC} \)

\[\text{Algorithm 1: Conical Compare-and-Capture Algorithm}\]

1. Select \( x_C \in \left[\frac{r - \rho r}{2\theta v_{xC}}, \min\{r + \rho, \frac{1 - \rho r}{1 + r} \}\right] \).
2. Wait until time \( 1 - \min\{1, x_C + r + 2\theta v_{xC}\} \).
3. For each epoch \( k \geq 1 \) do
   4. If \(|S_{\text{left}}^k| < |S_{\text{right}}^k|\) then
      5. Set \( \beta = 0, \bar{\beta} = \theta \)
      6. Move on angular path to location \((x_C, \theta)\)
      7. Move on angular path to return to \((x_C, 0)\)
   8. Else
      9. Set \( \beta = -\theta, \bar{\beta} = 0 \)
     10. Move on angular path to location \((x_C, -\theta)\)
     11. Move on angular path to return to \((x_C, 0)\)
   12. End

**C. Stay Near Perimeter (SNP) Algorithm**

Unlike the previous two algorithms, in this algorithm, the vehicle does not follow an angular path. Instead, the idea is to partition the environment into sectors and position the vehicle close to the perimeter in a specific sector.

We partition the environment \( \mathcal{E}(\theta) \) into \( n_s = \left\lfloor \frac{\theta}{\beta} \right\rfloor \) sectors, each with angle \( 2\theta_s = 2\arctan\left(\frac{\beta}{\rho}\right) \) (Fig. 3). Since \( r < \rho \), \( \theta_s < \frac{\pi}{4} \). Let \( N_i, l \in \{1, \ldots, n_s\} \) denote the \( l \)-th sector, where \( N_1 \) corresponds to the leftmost sector in the environment. Then, a resting point \((x_i, \alpha_i) \in \mathcal{E}(\theta)\) of a sector \( N_i \) is defined as the location for the vehicle such that when positioned at that location, the portion of the perimeter within that sector is contained completely within the capture radius of the vehicle. Mathematically, the resting point \((x_i, \alpha_i)\), for a sector \( N_i \) is defined as \( (\frac{\rho}{\cos(\alpha_i)}, (l - \frac{n_s + 1}{2})\theta_s) \). Further, we define \( D \) as the distance between the two resting points that are farthest in the environment (Fig. 3) as

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

If there is only one sector, i.e., \( n_s = 1 \Rightarrow D = 0 \) then this
implies the capture circle can contain the entire perimeter. Thus, by positioning the vehicle at the unique corresponding resting point, the vehicle can capture all intruders that arrive in the environment.

After partitioning the environment into $n_v$ sectors, SNP divides the environment into three annuli with width equal to $Dv$ each. This is equivalent to dividing time into intervals of duration $D$ each. Specifically, the $j$th time interval for any $j > 0$ is defined as the time interval $[(j - 1)D, jD]$ (Fig. \ref{fig:sector}).

To ensure a finite competitiveness, we require $\max \{x_{i,0}, |x_{i,1}|\} \leq l \rho$ for any $i \in \{1, \ldots, n_v\}$, i.e., the intruders require at least $3D$ time to reach the perimeter. For any $j \geq 1$, let $S_j^{\ast}$ be the set of intruders that arrive in a sector $N_i$ in the $j$th interval (Fig. \ref{fig:sector}).

The SNP algorithm (defined in Algorithm \ref{alg:SNP}) is based on the following two steps: First, select a sector in the environment with 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.

In the first comparison C1 (Line 6 in Algorithm \ref{alg:SNP}), SNP determines the 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 vehicle is presently located. In particular, suppose that the vehicle is located at the resting point of sector $N_i$ at the $j$th iteration. Corresponding to any sector $N_i$, we define $\eta_i$ as $|S_i^{j+2}| + |S_i^{j+3}|$ if $l \neq i$ and $|S_i^{j+1}| + |S_i^{j+2}| + |S_i^{j+3}|$, otherwise. Then, SNP selects the sector $N_{k^*}$, where $k^* = \arg \max_{k \in \{1, \ldots, n_v\}} \{\eta_1, \ldots, \eta_{n_v}\}$. In case there are multiple sectors with same number of intruders, then SNP breaks the tie as follows. If the tie includes the sector $N_i$, then SNP selects $N_i$.

Otherwise, 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 selecting the sector with the least index. Let the sector chosen as the outcome of C1 be $N_o, o \in \{1, \ldots, n_v\}$.

For the second comparison C2 (Line 7), if the sector obtained from C1 is $N_o$ with $o \neq i$, and the total number of intruders in the set $S_i^{j+2}$ is at least the total number of intruders in $S_i^{j+1}$, then SNP moves (Line 8) the vehicle to the resting point of sector $N_o$, denoted by $(x_{i,0}, \alpha_i)$, arriving there in at most $D$ time units. Then, the vehicle waits at $(x_{i,0}, \alpha_i)$ to capture all intruders in $S_i^{j+2}$. Otherwise, i.e., if $S_i^{j+2} < S_i^{j+1}$ or $o = i$), the vehicle stays (Line 10) at its current location $(x_i, \alpha_i)$, captures intruders in $S_i^{j+1}$ and then reevaluates after $D$ time units.

At time $t$, the vehicle waits for $D$ time units at location $(0, 0)$ after the first intruder arrives in the environment. Then the vehicle moves to the sector which has the maximum number of intruders in $S_i^1$, \forall $N_i$ sectors in the environment (Line 2). The vehicle then waits until time $3D$. To ensure that no intruder is lost until time $3D$, we require $\frac{D}{\cos(\pi/2)} \leq 2D$.

**Lemma IV.4** Let the vehicle be located at a resting point $(x_i, \alpha_i)$ of a sector $N_i, i \in \{1, \ldots, n_v\}$. Then, for any $j \geq 1$, the vehicle always captures intruders in either $S_i^{j+1}$ or $S_i^{j+2}$, where $N_o$ denotes the sector selected by SNP after C1.

**Proof:** Consider that the sector $N_o = N_i$. Then, according to Algorithm \ref{alg:SNP}, the vehicle stays at its current position and captures $S_i^{j+1}$ and the result follows.

Now consider that the sector $N_o \neq N_i$. Then there are two cases: (i) Either the vehicle decides to stay at its current position for $D$ time interval, i.e., $|S_i^{j+1}| > |S_i^{j+2}|$ or (ii) the vehicle decides to move to the resting point corresponding to the sector $N_o$, i.e., $|S_i^{j+1}| \leq |S_i^{j+2}|$. In case (i), the vehicle stays at its current location and captures $|S_i^{j+1}|$. In case (ii), the vehicle spends at most $D$ time units to move to the resting point of the sector $N_o$ and then captures intruders in the set $S_i^{j+2}$. This concludes the proof.

To establish the competitive ratio of Algorithm 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 lemmas will jointly establish the competitive ratio of SNP algorithm.

**Lemma IV.5** In algorithm SNP, any two consecutive captured intervals pay for a total of $3(n_v - 1)$ lost intervals.

**Proof:** As Lemma IV.4 ensures that the vehicle always captures an interval of intruders, any two consecutive captured intervals can be classified into four types (see [21] for images); (a) stay at the current location and capture both intervals on the same side, (b) stay at the current location and capture an interval and then move to the resting point of $N_o$ and capture the second interval, (c) move to the resting point of $N_o$ and capture both intervals, and finally (d) move to the resting point of sector $N_o$ and capture an interval and then move to the resting point of another sector, $N_o', o' \in \{1, \ldots, n_v\} \setminus \{o\}$ and capture an interval.

The explanation for Type (a) captured intervals $S_i^{j+1} + S_i^{j+2}$ is as follows. At time instant $jD$ and $(j + 1)D$, since vehicle decides to capture $S_i^{j+1}$ and $S_i^{j+2}$ (comparison C1 and C2), it loses $S_i^{j+2}$ and $S_i^{j+3}$ intruders from other sectors, i.e., \forall $1 \in \{1, \ldots, n_v\} \setminus \{i\}$. Thus the captured intervals $S_i^{j+1}$ and $S_i^{j+2}$ are charged $2n_v - 2$ times. The remaining $n_v - 1$ charge is explained as follows. Since the vehicle is currently located at $(x_i, \alpha_i)$ it must be that the vehicle captured $S_i^1$. This implies that comparison C1 must have yielded sector $N_i$ at either time instant $(j - 2)D$ (if the vehicle was located at $(x_i, \alpha_i), t \neq i))$ or $(j - 1)D$ (if the vehicle was located

---

**Algorithm 2: Stay Near Perimeter (SNP) Algorithm**

1. Stay at origin until time $D$.
2. $k^* = \arg \max_{k \in \{1, \ldots, n_v\}} \{\eta_1, \ldots, \eta_{n_v}\}$, $N_i = N_{k^*}$.
3. Move to $(x_i, \alpha_i)$ and wait until time $3D$.
4. Assumes vehicle is at $(x_i, \alpha_i)$ in sector $N_i$.
5. For each $j \geq 1$ do
   6. $k^* = \arg \max_{k \in \{1, \ldots, n_v\}} \{\eta_1, \ldots, \eta_{n_v}\}$, $N_o = N_{k^*}$.
   7. If $N_o \neq N_i$ and $|S_i^{j+2}| \geq |S_i^{j+1}|$ then
      8. Move to $(x_o, \alpha_o)$ and then capture $|S_i^{j+2}|$
    9. Else
      10. Stay at $(x_i, \alpha_i)$ and capture $|S_i^{j+1}|$
    11. End
12. End
The final pay is the remaining pay captured by the combination of one or more traces as described in Lemma IV.6.

For type (d) captured intervals, without loss of generality, consider that after capturing its first interval, $S_{i+1}^j$ in sector $N_o$, the vehicle moves back to sector $N_i$ to capture its second interval $S_{i+2}^j$, i.e., $N_o = N_i$. Type (d) captured interval $S_{i+2}^j$ pays once for $S_{i+1}^j$, $S_{i+2}^j$, and $S_{i+3}^j$ combined, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ (comparison C1 and C2 at time $j+1$). The captured intervals also pay $n_x - 1$ times for lost intervals $S_{i+2}^j$, for all $N_i, l \neq o$ sectors. Finally, the last $n_x - 1$ pays is for lost intervals $S_{i+2}^j$ and $S_{i+1}^j$, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ as the vehicle captured $S_{i+2}^j$ instead of $S_{i+1}^j$ (comparison C1).

For type (d) captured intervals, without loss of generality, consider that after capturing its first interval, $S_{i+1}^j$ in sector $N_o$, the vehicle captured and combined, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ (comparison C1 and C2 at time $j+1$). The captured intervals also pay $n_x - 1$ times for lost intervals $S_{i+2}^j$, for all $N_i, l \neq o$ sectors. Finally, the last $n_x - 1$ pays is for lost intervals $S_{i+2}^j$ and $S_{i+1}^j$, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ as the vehicle captured $S_{i+2}^j$ instead of $S_{i+1}^j$ (comparison C1).

Since each type of captured intervals are charged $3(n_x-1)$ times, the result is established.

We now establish that each lost interval is fully accounted for by the captured intervals. Since SNP directs the vehicle to stay at a resting point of any sector for some time interval, it can be viewed as a sequence of traces, in which the vehicle spends some number of intervals at one resting point and some number of intervals at another. Each trace is thus defined by a set \{k_1, k_2, \ldots, k_n\}, where each element $k_l, l \in \{1, \ldots, n_o\}$ denotes the number of intervals that the vehicle decides to capture by staying at the corresponding resting point of the sector $N_l$.

Lemma IV.6 Each lost interval is accounted for by the captured intervals of SNP algorithm.

Proof: Note that any realization of SNP can be achieved by the combination of one or more traces as described in the following cases. Case (i) $k_l = 3$ and $k_l = 0 \forall l \in \{1, \ldots, n_o\} \setminus \{i\}$. Case (ii) $0 \leq k_l < 3$ and $k_o = 2$ and Case (iii) $k_l = 0, k_o = 1$ and $k_o = 1, \forall o \in \{1, \ldots, n_o\} \setminus \{i\}$ and $\forall o' \in \{1, \ldots, n_o\} \setminus \{i\}$. 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 $S_{i+1}^j, S_{i+2}^j$, and $S_{i+3}^j$ account for all of the lost intervals $S_{i+2}^j$ and $S_{i+3}^j, \forall l \in \{1, \ldots, n_o\} \setminus \{i\}$. 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 $|S_{i+2}^j| < |S_{i+2}^j|$ (comparison C2). We first consider sub-case (a). Sub-case (a) implies that at time instant $jD$, the total number of intruders in sector $N_i$ is more than in any other sector in the environment. Thus, captured intervals $S_{i+1}^j, S_{i+2}^j$, and $S_{i+3}^j$ account for all of the lost intervals $S_{i+2}^j$ and $S_{i+3}^j, \forall l \neq i$. In sub-case (b), we account for lost intervals $S_{i+2}^j, S_{i+3}^j, \forall l \in \{1, \ldots, n_o\} \setminus \{i, o\}$ and $S_{i+2}^j, S_{i+3}^j$, separately. Lost intervals $S_{i+2}^j$ and $S_{i+3}^j$ are accounted for because $|S_{i+2}^j| + |S_{i+3}^j| \leq |S_{i+1}^j| + |S_{i+2}^j| + |S_{i+3}^j|$ or equivalently $\eta_i^j \leq \eta_i^j$ (comparison C1). Now it remains to account for lost intervals $S_{i+2}^j$ and $S_{i+3}^j$. Observe that if there exists a sector $N_o \neq N_i$ at time instant $jD$ such that $|S_{i+2}^j| < |S_{i+1}^j|$, then there cannot exist the same $N_o$ at time instant $(j+1)D$ (from comparison C1). Thus, even if $N_o \neq N_i$, exists, then the lost interval $S_{i+2}^j$ is accounted by $S_{i+1}^j$ as $|S_{i+2}^j| < |S_{i+1}^j|$ (comparison C2). Since, at time $(j+1)D$, sector $N_o$ cannot be selected again, it follows that $\eta_i^j < \eta_i^j$ at time $(j+1)D$ and thus, $S_{i+3}^j$ is accounted for.

Case (ii): To account for the lost intervals $S_{i+1}^{j+1}$ and $S_{i+2}^{j+1}$, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$, from comparison C1 and C2 at time $(j+1)D$, the vehicle was supposed to capture all $S_{i+2}^{j-1} + k, S_{i+1}^{j-1} + k, \ldots, S_{i+1}^{j+k}$ intervals. While the vehicle captured $S_{i+1}^{j-2} + k, S_{i+1}^{j-2} + k, \ldots, S_{i+1}^{j+k}$ intervals, it did not capture $S_{i+1}^{j+k}$. As $\eta_i^j > \eta_i^j$ at time instant $(j+k)D$, lost intervals $S_{i+1}^{j+k}$, $\forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ are fully accounted for. The remaining lost intervals $S_{i+2}^{j+k}, S_{i+2}^{j+k}, S_{i+2}^{j+k}, S_{i+2}^{j+k}, \forall l \in \{1, \ldots, n_o\} \setminus \{i\}$ are fully accounted by the captured intervals $S_{i+2}^{j+k}$ and $S_{i+2}^{j+k}$, because the conditions $\eta_i^j > \eta_i^j$ and $\eta_o^j > \eta_i^j$ are satisfied at time instant $(j+k)D$ (comparison C1).

Case (iii): To account for lost intervals $S_{i+1}^{j+k}, S_{i+1}^{j+k}, S_{i+1}^{j+k}$, and $S_{i+1}^{j+k}, S_{i+2}^{j+k}$, and $S_{i+2}^{j+k}$, $\forall l \in \{1, \ldots, n_o\} \setminus \{i, o\}$, the vehicle was supposed to capture $S_{i+1}^{j+k}$ and $S_{i+1}^{j+k}$. This follows because at time instant $jD$, $\eta_i^j > \eta_i^j$ (comparison C1) and $|S_{i+1}^{j+k}| \geq |S_{i+1}^{j+k}|$ (comparison C2). The vehicle captured $S_{i+2}^{j+k}$ which accounts for $S_{i+1}^{j+k}$ as $|S_{i+1}^{j+k}| \geq |S_{i+1}^{j+k}|$. As the vehicle moved to capture $S_{i+1}^{j+k}$ at time $(j+2)D$, it implies that $|S_{i+1}^{j+k}| \geq |S_{i+1}^{j+k}|$ (comparison C2) and thus, $S_{i+1}^{j+k}, S_{i+2}^{j+k}, S_{i+2}^{j+k}, S_{i+2}^{j+k}$ are all accounted by the captured interval $|S_{i+1}^{j+k}|$.

Finally, the lost intervals $|S_{i+1}^{j+k}|, \forall l \in \{1, \ldots, n_o\} \setminus \{i, o\}$ are accounted for as follows: If the vehicle also captures $S_{i+1}^{j+k}$, then lost intervals $S_{i+1}^{j+k}$ are accounted for by case (ii) ($k_l = 1$). Otherwise (i.e., the vehicle moved to another sector $N_o, \delta \neq o$ to capture $S_{i+1}^{j+k}$, $S_{i+1}^{j+k}$ is accounted for as per case (iii) as now the lost intervals will be $S_{i+1}^{j+k}, S_{i+1}^{j+k}, S_{i+1}^{j+k}, S_{i+1}^{j+k}, S_{i+1}^{j+k}$, and $S_{i+1}^{j+k}$ are all accounted by the captured interval $S_{i+1}^{j+k}$). Finally, note that the boundary cases of the first and the last intervals fall into these cases by adding dummy intervals $S_0, \forall l \in \{1, \ldots, n_o\}$ and $S_{i+1}^{j+k}$, where $Y$ denotes the last interval that consists of intruders in any sector, each with zero cardinality. We assume that the vehicle captures all of the dummy intervals. This concludes the proof.
Theorem IV.7 (SNP competitiveness) For any problem instance \( \mathcal{P}(\theta, r, v) \) that satisfies \( 3D \leq \frac{1-v}{v} \) and \( \frac{2}{\rho \cos(\theta_s)} \leq 2D \), SNP is \( \frac{3n_s-1}{2} \)-competitive, where \( n_s = \lceil \theta/\theta_s \rceil \), \( \theta_s = \arctan(r/\rho) \) and \( D \) is defined in (3).

Proof: From Lemma IV.5 and Lemma IV.6 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. Assuming that the optimal offline algorithm captures all intruder intervals, i.e., \( 3n_s-1 \), the claim follows.

V. NUMERICAL VISUALIZATION AND OBSERVATIONS

We now provide a numerical visualization of the analytic bounds derived in this paper. Figure 4 shows the \((\rho, v)\) parameter regime plot for a fixed capture radius \( r = 0.05 \) and \( \theta = \frac{\pi}{3} \). We have provided additional parameter regime plots for different values of \( r \) and \( \theta \) in [21].

Since the competitiveness of SNP depends on the number of sectors, we observe that the parameter regime of SNP is in regions, where each region corresponds to a specific competitiveness. As the capture radius \( r \) increases or the angle \( \theta \) decreases, the number of regions decreases. An important characteristic for SNP is that it can be used to determine the tradeoff between the competitiveness and the target parameter regime for the problem instance.

Figure 4 suggests that for small values of \( r \), SNP has a relatively large region of utility implying that the smaller the capture radius, SNP can capture equally fast intruders, but at the cost of higher competitive ratio. For \( r = 0.05 \), SNP is 2.5-competitive for \( \rho < 0.2 \). Interestingly, the curve for SNP extends beyond that of Theorem III.4. We observe that for high values of \( \rho \), the curve defined by sufficient conditions for SNP is completely below the curve defined by conditions of ConCaC suggesting that SNP is ineffective for large \( \rho \). A similar observation is made for high values of \( r \).

VI. CONCLUSION AND FUTURE DIRECTIONS

This work analyzed the problem wherein a single vehicle, having a finite capture radius, is tasked to defend a perimeter in a conical environment from arbitrary many intruders that arrive in the environment in an arbitrary fashion. We designed and analyzed three algorithms and established sufficient conditions that guarantee a finite competitive ratio for each algorithm. As there is a trade-off in covering a larger parameter regime and achieving a smaller competitive ratio, the choice of which algorithm to use depends on the problem parameters and the acceptable bound on competitiveness. We also derived two fundamental limits on achieving a finite competitive ratio by any online algorithm.

Key future directions include a cooperative multi-vehicle scenario with communication and energy constraints.

REFERENCES

[1] A. Von Moll, E. Garcia, D. Casbeer, M. Suresh, and S. C. Swar, “Multiple-pursuer, single-evader border defense differential game,” Journal of Aerospace Info. Systems, vol. 17, no. 8, pp. 407–416, 2020.
[2] D. G. Macharet, A. K. Chen, D. Shishika, G. J. Pappas, and V. Kumar, “Adaptive partitioning for coordinated multi-agent perimeter defense,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 7971–7977.
[3] S. Bajaj and S. D. Bopadikar, “Dynamic boundary guarding against radially incoming targets,” in 2019 IEEE 58th Conference on Decision and Control (CDC), 2019, pp. 4804–4809.
[4] A. Von Moll, D. Shishika, Z. Fuchs, and M. Dorothy, “The turret-runner-penetrator differential game,” in 2021 American Control Conference (ACC). IEEE, 2021, pp. 3202–3209.
[5] D. D. Sleator and R. E. Tarjan, “Amortized efficiency of list update and paging rules,” Comm. of ACM, vol. 28, no. 2, pp. 202–208, 1985.
[6] H. N. Purashagis, “Dynamic vehicle routing problems,” Vehicle routing: Methods and studies, vol. 16, pp. 223–248, 1988.
[7] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, “Dynamic vehicle routing for robotic systems,” Proceedings of the IEEE, vol. 99, no. 9, pp. 1482–1504, 2011.
[8] S. L. Smith, S. D. Bopadikar, and F. Bullo, “A dynamic boundary guarding problem with translating targets,” in Proceedings of the 48th IEEE Conference on Decision and Control (CDC), 2019, pp. 8543–8548.
[9] R. Isaacs, “Differential games. a mathematical theory with applications to warfare and pursuit, control and optimization,” 1965.
[10] M. Chen, Z. Zhou, and C. J. Tomlin, “Multiplayer reach-avoid games via pairwise outcomes,” IEEE Transactions on Automatic Control, vol. 62, no. 3, pp. 1451–1457, 2016.
[11] E. Garcia, A. Von Moll, D. W. Casbeer, and M. Pachter, “Strategies for defending a coastline against multiple attackers,” in 2019 IEEE 58th CDC19. IEEE, 2019, pp. 7319–7324.
[12] A. Davydov, P. Rivera-Ortiz, and Y. Diaz-Mercado, “Pursuer coordination in multi-player reach-avoid games through control barrier functions,” IEEE Control Systems Letters, vol. 5, no. 6, pp. 1910–1915, 2020.
[13] K. Margellos and J. Lygeros, “Hamilton–Jacobi formulation for reach–avoid differential games,” IEEE Transactions on Automatic Control, vol. 56, no. 8, pp. 1849–1861, 2011.
[14] M. Chen, Z. Zhou, and C. J. Tomlin, “A path defense approach to the multiplayer reach-avoid game,” in 53rd IEEE Conference on Decision and Control. IEEE, 2014, pp. 2420–2426.
[15] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, “Matching-based capture strategies for 3d heterogeneous multiplayer reach-avoid differential games,” Automatica, vol. 140, p. 110207, 2022.
[16] D. Shishika and V. Kumar, “Perimeter-defense game on arbitrary convex shapes,” arXiv preprint, arXiv:1906.03999, 2019.
[17] S. Velhal, S. Sundaram, and N. Sundararajan, “A decentralized multi-uv spatio-temporal multi-task allocation approach for perimeter defense,” arXiv preprint, arXiv:2102.07381, 2021.
[18] Y. Lee and B. Bakolas, “Guarding a convex target set from an attacker in Euclidean spaces,” IEEE Control Systems Letters, vol. 6, pp. 1706–1711, 2021.
[19] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, “Matching-based capture strategies for 3d heterogeneous multiplayer reach-avoid differential games,” 2019, online available at:https://arxiv.org/abs/1909.11881.
[20] S. Bajaj, E. Torng, and S. D. Bopadikar, “Competitive perimeter defense on a line,” in 2021 American Control Conference (ACC), 2021, pp. 3196–3201.
[21] S. Bajaj, E. Torng, S. D. Bopadikar, A. Von Moll, I. Weintraub, E. Garcia, and D. W. Casbeer, “Competitive perimeter defense of conical environments,” arXiv preprint arXiv:2110.04667v2, 2021, online available at: https://arxiv.org/abs/2110.04667v2.