Competitive Perimeter Defense on a Line

Shivam Bajaj[1] Eric Torng[2] Shaunak D. Bopardikar[1]

***Abstract*— We consider a perimeter defense problem in which a single vehicle seeks to defend a compact region from intruders in a one-dimensional environment parameterized by the perimeter size and the intruder-to-vehicle speed ratio. The intruders move inward with fixed speed and direction to reach the perimeter. We provide both positive and negative *worst-case* performance results over the parameter space using competitive analysis. We first establish fundamental limits by identifying the most difficult parameter combinations that admit no c-competitive algorithms for any constant $c \geq 1$ and slightly easier parameter combinations in which every algorithm is at best 2-competitive. We then design three classes of algorithms and prove they are 1, 2, and 4-competitive, respectively, for increasingly difficult parameter combinations. Finally, we present numerical studies that provide insights into the performance of these algorithms against stochastically generated intruders.**

I. INTRODUCTION

This paper addresses a perimeter defense problem, which is a class of vehicle routing problems, in which a vehicle seeks to intercept mobile intruders before they reach a specified region. In our problem, a robotic vehicle must defend a subset of a line segment from intruders that are generated at the endpoints of the line segment and move towards the subset, with a fixed speed. The robotic defender moves with maximum unit speed with the goal of *capturing* the maximum number of intruders. This perimeter defense problem is an *online* problem in that the input, consisting of the arrival of intruders at specified times and locations, is only revealed gradually over time.

While perimeter defense problems have been well-studied, most prior work has focused on determining an optimal strategy for a small number of intruders or assuming that the input instance is generated by some stochastic process. While these results provide valuable insights into the average-case performance of defense strategies, they essentially ignore the worst-case where intruders may coordinate their actions to overwhelm the defense.

To understand how algorithms that specify vehicle motion perform in the worst-case, we adopt a competitive analysis technique [1]. In competitive analysis, we measure the performance of an online algorithm A using the concept of *competitive ratio*, i.e., the ratio of an optimal offline algorithm's performance divided by algorithm A's performance for a worst-case input instance. Algorithm A 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 instances.

The primary application for our work is defending a perimeter from intruders such as missiles or locusts. Additional applications include gathering information on mobile entities in surveillance or traffic scenarios.

Perimeter defense problems were first introduced for a single vehicle and a single intruder in [2]. Since then, perimeter defense has been mostly formulated as a pursuit-evasion 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 [3], [4]. A typical approach requires computing solutions to the Hamilton-Jacobi-Bellman-Isaacs equation which is suitable for low dimensional state spaces and in simple environments [5], [6]. Recent works include [7], which proposes a receding horizon strategy based on maximum matching, and [8], which considers a scenario where the defenders are constrained to be on the perimeter.

In vehicle routing problems, inputs become available over time. Introduced on graphs in [9], a typical approach requires that the vehicle routes be re-planned as new information is revealed over time [10]. The inputs may have multiple levels of priorities [11] or can be randomly recalled [12]. We refer the reader to [13] for a review of this literature. A common way to analyze the performance of online algorithms is competitive analysis [14], [15], [16].

Another area of related work is the class of Moving Target Travelling Salesman Problem (MTTSP) on a single line [17], [18]. Several variants of this problem are discussed in [19]. Specifically, the authors provide an $\mathcal{O}(n^2)$ algorithm to capture n intruders on a line in minimum time.

Earlier, we introduced a perimeter defense problem for a circular and rectangular environment with stochastically generated input [20], [21]. The key distinction of the current work from these past works is the characterization of *competitiveness* of the algorithms for worst case inputs.

Our general contribution is the following: we introduce a perimeter defense problem against mobile intruders using competitive analysis to derive worst-case performance guarantees. Specifically, we consider an environment comprising a line segment $[-1, 1]$ in which the intruders arrive as per an arbitrary sequence at the endpoints. After arrival, the intruders move toward the origin, with fixed speed $v < 1$, with the objective of reaching the region $[-\rho, \rho]$ called the *perimeter*, for a given $0 < \rho < 1$. A vehicle, modelled as a first-order integrator with a maximum speed of unity, seeks

[1]Shivam Bajaj and Shaunak D. Bopardikar are with the Department of Electrical and Computer Engineering, Michigan State University. email: bajajshi@msu.edu. [2]Eric Torng is with the Department of Computer Science and Engineering, Michigan State University.

This work has been supported in part by NSF Award ECCS-2030556.

to capture (become coincident with) the intruders before they reach the perimeter. Our specific contributions are as follows. We first characterize a most difficult parameter regime in $v-\rho$ space in which no control algorithm (either online or offline) for the vehicle can be c-competitive for any constant c and a second, slightly easier, parameter regime in which no algorithm is better than 2-competitive. We also show that a class of simple algorithms, such as the First-Come-First-Served, are not c-competitive, even for parameter regimes in which other algorithms are. Next, we design and analyze three algorithms establishing 1, 2, and 4-competitiveness, respectively, for increasingly difficult parameter regimes. We numerically characterize their performance when the intruders are generated as per a stochastic process. We observe that the algorithms capture at least half the intruders generated even for parameter settings beyond their respective parameter regime.

The paper is organized as follows. In Section II, we formally define our problem and the competitive ratio. We derive fundamental limits on how competitive any algorithm can be for difficult parameter regimes in Section III. In Section IV, we design and analyze three algorithms. Section V presents numerical simulations. In Section VI, we present a summary and possible directions for future work.

II. Problem Formulation

Consider an environment $\mathcal{E}(\rho)=\{y\in\mathbb{R}\,:\,\rho\le|y|\le 1\}$ and let $\bar{\mathcal{E}}:=[-1,1]$. Intruders arrive over time at either location -1 or 1 and move, with fixed speed $v<1$, towards the nearest point in $\bar{\mathcal{E}}$ out of $-\rho$ or ρ. The defense consists of a single vehicle with motion modeled as a first order integrator. The vehicle can move with a maximum speed of unity. 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. Let $n(t)$ denote the number of intruders arrived at time instant t. An input instance I is a tuple comprising the time instants, the corresponding number of intruders generated at those instants and their corresponding initial locations, is defined as $I=\{t,n(t),\{-1,1\}^{n(t)}\}_{t=0}^{T}$.

An online algorithm determines the velocity for the vehicle as a function of the location of the intruders that have arrived in the environment up to the current time instant t. Let $\mathcal{Q}(t,I)$ denote the set of instantaneous locations of all intruders in the environment at time t from the input instance I. An intruder is removed from $\mathcal{Q}(t,I)$ if it is captured or lost. We now formally define an online algorithm as follows.

Online Algorithm: An online algorithm for a vehicle is a map $\mathcal{A}:\bar{\mathcal{E}}\times\mathbb{F}\rightarrow[-1,1]$, where $\mathbb{F}(\mathcal{E})$ is the set of finite subsets of $\mathcal{E}$, assigning a commanded velocity to the vehicle as a function of the position of the vehicle, denoted as $x(t)$, and the location of the intruders, yielding the kinematic model, $\dot{x}(t)=\mathcal{A}(x(t),\mathcal{Q}(t,I))$.

An optimal *offline algorithm* is a non-causal algorithm having complete information of the input instance I at any time $t\le T$ and thus, the velocity of the vehicle is a function of current and future locations of the intruders.

Definition 1 (Competitive Ratio) *Given an $\mathcal{E}(\rho)$, an input instance I for $\mathcal{E}(\rho)$, and a given online algorithm $\mathcal{A}$, let $n_{\mathcal{A}}(I)$ denote the number of intruders captured by the vehicle when following algorithm $\mathcal{A}$ on input instance I. Let OPT denote the optimal algorithm that maximizes the number of intruders captured out of input instance I. Then, the competitive ratio of $\mathcal{A}$ on I is defined as $c_{\mathcal{A}}(I)=n_{OPT}(I)/n_{\mathcal{A}}(I)\ge 1$, and the competitive ratio of $\mathcal{A}$ for environment $\mathcal{E}$ is $c_{\mathcal{A}}(\mathcal{E})=\sup_I c_{\mathcal{A}}(I)$. Finally, the competitive ratio for environment $\mathcal{E}$ is $c(\mathcal{E})=\inf_{\mathcal{A}} c_{\mathcal{A}}(\mathcal{E})$. We say that an algorithm is c-competitive for $\mathcal{E}$ if $c_{\mathcal{A}}(\mathcal{E})\le c$.*

We assume that all of the input instances are non-adaptive where the arrival of intruders is not based on the movement of the vehicle.

Problem Statement: The aim of this paper is to design c-competitive algorithms for the vehicle with minimum c.

In what follows, we provide only the outline of the proofs for brevity. The detailed proofs are contained in [24].

III. Fundamental Limits

We begin by characterizing a property of an *extreme speed algorithm*, i.e., $\mathcal{A}':\bar{\mathcal{E}}\times\mathbb{F}\rightarrow\{-1,0,1\}$, which either moves the vehicle with unit speed or keeps it stationary.

Lemma 1 (Extreme speed algorithms) *Given an arbitrary algorithm $\mathcal{A}$ and a non-adaptive input instance I, there exists an extreme speed algorithm $\mathcal{A}'$ that captures at least as many intruders in I as $\mathcal{A}$.*

Proof: [Outline] We define the *capture profile* of $\mathcal{A}$'s execution on an arbitrary, non-adaptive input instance I as the set of pairs (x_i,k_i) for $1\le i\le n$ where k_i is the time of the ith capture and x_i is the point where the ith capture occurred. The crux of this proof is showing that we can create an extreme speed algorithm $\mathcal{A}'$ that has the same capture profile as $\mathcal{A}$ by getting to the next capture point x_i as quickly as possible and, if necessary, waiting at location x_i until time k_i. ■

In light of Lemma 1, we can restrict our attention to algorithms that either move the vehicle with maximum speed or keep the vehicle at rest.

We now present the fundamental limits. We first present a fundamental limit on achieving a c-competitive ratio for any constant c followed by a fundamental limit on achieving a 2-competitive ratio.

Theorem III.1 *For any environment $\mathcal{E}$ such that $v>\frac{1-\rho}{2\rho}$,*

i) *there does not exist a c-competitive algorithm and*
ii) *no algorithm (online/offline) can capture all intruders.*

Proof: [Outline] The idea behind the proof is to first construct a worst case input for any online algorithm and then compare the performance with the optimal algorithm. The input instance consists of two phases: a stream of intruders that arrive at the endpoint 1, 2 time units apart starting at time 1, and a burst of $c+1$ intruders who arrive at the endpoint -1

at time t that corresponds to the first time the vehicle moves to ρ according to any online algorithm. Then, for $v > \frac{1-\rho}{2\rho}$, any online algorithm captures at most 1 stream intruder and loses the entire burst of $c+1$ intruders, however, the optimal algorithm can capture all the burst intruders and at least all but the last stream intruder. ■

The following theorem provides a fundamental limit on achieving a 2-competitive ratio for any environment.

Theorem III.2 *For the environment $\mathcal{E}$ such that $v \geq \frac{1-\rho}{1+\rho}$, $c(\mathcal{E}) \geq 2$.*

Proof: [Outline] In this proof, we consider five different input instances consisting of two intruders, a and b, arriving at opposite endpoints. We show that no single online algorithm can capture both intruders from all five input instances. In contrast, we then show there exists a collection of algorithms, one of which captures both intruders for each of the five input instances. ■

We now show that a natural algorithm, *First-Come-First-Served* (FCFS), cannot be a competitive algorithm for this problem for a sufficiently difficult parameter regime. We define FCFS as the algorithm which sends the vehicle with speed 1 towards the earliest intruder to arrive that is not lost or already captured.

Lemma 2 *For any $\mathcal{E}$ 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: [Outline] We prove this result by constructing a worst case input instance $I(c)$ against FCFS. Let the first intruder be released at time 0 at 1. Let $c+1$ intruders be released at time ϵ at -1. Since, $\frac{2}{v+1} + \rho > \frac{1-\rho}{v} + \epsilon$, it follows that $FCFS(I(c)) = 1$, where $A(I)$ is the number of intruders captured by an algorithm A in an input instance I. On the other hand, since the optimal algorithm has the information of the entire input instance, $OPT(I(c)) = c+1$. This result generalizes to any variation of FCFS which captures the first arriving intruder, if possible, before any later arriving intruder. ■

We now turn our attention to the design of algorithms with provable guarantees on the competitive ratio. In the following section, we describe and analyze three algorithms, characterizing parameter regimes with provably finite competitive ratios.

IV. ALGORITHMS

We now propose three main algorithms for the vehicle that are provably 1, 2, and 4 competitive. As the competitive ratio increases, the parameter regime that can be handled also increases.

A. Sweeping algorithm

We define the Sweeping algorithm (Sweep) as follows. At time 0, the vehicle moves with unit speed toward endpoint $+1$. From this point on, the vehicle only changes direction when it reaches an endpoint $+1$ or -1, at which time it moves with unit speed towards the opposite endpoint. Sweep is an *open-loop* algorithm; that is, it ignores all information about intruders. One logical variant is to stop moving in a given direction if there are no intruders in that direction. We show this variant achieves the same performance guarantee.

Theorem IV.1 *For environment $\mathcal{E}$, Sweep is 1-competitive if $v \leq \frac{1-\rho}{3+\rho}$. If not, it is not c-competitive for any constant c.*

Proof: Suppose that $v \leq \frac{1-\rho}{3+\rho}$ holds. We show Sweep captures all intruders. Any intruder i will take $\frac{1-\rho}{v}$ time to get from its arrival location, which we now assume to be 1 without loss of generality, to ρ. In the worst case for Sweep, which is that it has left 1 just before intruder i arrived, it will take $3+\rho$ time to get to ρ moving towards intruder i. If $3+\rho \leq \frac{1-\rho}{v}$, then vehicle will get to ρ first and intruder i will be captured, and the first result follows. For $v > \frac{1-\rho}{3+\rho}$ there is an input instance where intruders only arrive at 1 just after the vehicle has left 1. As $v > \frac{1-\rho}{3+\rho}$, all intruders will be lost and the second result follows. ■

We observe that the upper bound in the proof holds for the Sweep variant that stops moving in a given direction if there are no intruders in that direction. The lower bound requires introducing some intruders at -1 to ensure that the Sweep variant will move the vehicle towards -1. These intruders might be captured, but the lower bound still holds by increasing the number of intruders which arrive at 1.

B. Compare and Capture (CaC) 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., its actions depend only on the present state of the vehicle and the intruders.

We begin with some notation and definitions. An epoch k for the CaC algorithm is the time interval when the vehicle moves from location x_k to location x_{k+1} and is about to move from x_{k+1}, capturing some intruders along the way. Location x_k is always either ρ or $-\rho$. We denote the start of epoch k using the notation k_S. For epoch k, we define S^k_{same} as the set of intruders on the *same side* as the vehicle at time k_S, and S^k_{opp} as the set of intruders on the *opposite side* of the vehicle 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. Specifically, if the vehicle is located at $x_k = \rho$, then S^k_{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 $k \geq 1$, the algorithm computes the number of intruders located in S^k_{same} and S^k_{opp}. If the total number of intruders in S^k_{same} is greater than the total number of intruders in S^k_{opp}, then the vehicle moves away from the origin for at most $\frac{1-\rho}{1+v}$ time to capture all intruders from the set S^k_{same} and then returns to $x_{k+1} = x_k$. Otherwise, the vehicle moves for at most $2\rho + \frac{4v(1-\rho)}{(1+v)^2}$ time to capture all intruders located in S^k_{opp} and then returns to $x_{k+1} = -x_k$.

At time 0, we assume the vehicle starts at the origin. CaC waits at the origin until the first intruder that arrives in the

Algorithm 1: Compare-and-Capture Algorithm

1 Vehicle is at center and waits for $\frac{1-\rho-3\rho v}{v}$ time units
2 **if** *intruders in* $[\rho+3\rho v, 1] \leq$ *intruders in* $[-1, -(\rho+3\rho v)]$ **then**
3   Move to $-\rho$
4 **else**
5   Move to ρ.
6 **end**
7 **for** *each epoch* $k \geq 1$ **do**
8   **if** *at epoch* k, $|S^k_{\text{same}}| \leq |S^k_{\text{opp}}|$ **then**
9   Move to capture all intruders located in S^k_{opp}
10   Move to $x_{k+1} = -x_k$
11   **else**
12   Move to capture intruders located in S^k_{same}.
13   Move to $x_{k+1} = x_k$
14   **end**
15 **end**

environment is $3\rho v+\rho$ distance away from the origin, i.e., the vehicle does not move until $\frac{1-\rho-3\rho v}{v}$ time units after the first intruder arrives. If the total number of intruders located in $[\rho+3\rho v, 1]$ is greater than the total number of intruders located in $[-1, -(\rho+3\rho v)]$, then the vehicle moves to $x_1 = \rho$. If not, the vehicle moves to $x_1 = -\rho$. The first epoch begins when the vehicle reaches x_1.

To prove 2-competitiveness of CaC, we first prove that any intruder not belonging to S^k_{same} or S^k_{opp} in an epoch k will not be lost during epoch k.

Lemma 3 *In every epoch k, any intruder that lies outside of the set S^k_{same} and S^k_{opp} is not lost if $\frac{\rho v}{1-\rho} + \frac{v^2}{(1+v)^2} \leq \frac{1}{4}$.*

Proof: [Outline] We show that any intruder that did not belong to S^k_{same} and S^k_{opp} is at least $\rho+2\rho v$ distance away from the origin at the end of epoch k. ■

Lemma 4 *In every epoch k of the CaC, S^k_{opp} is well defined if $\rho+2\rho v+\frac{2v(1-\rho)}{1+v} \leq 1$.*

Proof: Omitted for brevity. ■

Theorem IV.2 *CaC captures at least half of all intruders and is 2-competitive in parameter regimes for which Lemma 3 and Lemma 4 both hold.*

Proof: Based on Lemma 4, S^k_{opp} is well-defined in every epoch k. This implies that Algorithm 1 is well-defined. Lemma 3 ensures that every intruder will belong to S^k_{same} or S^k_{opp} for some epoch k. Because of the comparison in line 9 of Algorithm 1, in every epoch, the number of intruders captured is at least the number of intruders lost. Note that an intruder is not lost in epoch k if it belongs to S^{k+1}_{same} or S^{k+1}_{opp} in the subsequent epoch $k+1$. Thus, the algorithm captures at least half of all intruders and is 2-competitive. ■

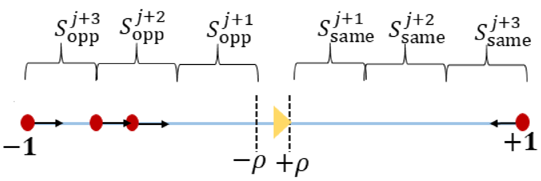

Fig. 1. Breakdown of the environment into regions of length $2\rho v$ by the CAP algorithm. The triangle depicts the vehicle, whereas a dot depicts an intruder

C. Capture with Patience (CAP) Algorithm

We now present another memoryless algorithm, Capture with Patience (CAP), in which the vehicle stays in the range $[-\rho, \rho]$ waiting for and capturing intruders at ρ or $-\rho$. CAP is only 4-competitive but can operate beyond the parameter regime of the CaC algorithm.

A key feature in environment $\mathcal{E}$ 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 ρ or $-\rho$. CAP requires that $z \geq 6\rho$, equivalently $v \leq \frac{1-\rho}{6\rho}$. This requirement ensures that incoming intruders take at least 6ρ time to get to either $-\rho$ or ρ whereas it takes the vehicle 2ρ time to move from $-\rho$ to ρ or vice versa.

CAP is formally defined in Algorithm 2 and is described as follows. First, to simplify notation, it defines time 0 to be the moment when the first intruder arrives. It then breaks time up into intervals of length 2ρ. The ith interval for $i \geq 1$ is defined as the time interval $[2(i-1)\rho, 2i\rho]$. We say that a set of intruders S is on the *same side* as the vehicle if the vehicle is located at ρ (resp. $-\rho$) and $S \subset (\rho, 1]$ (resp. $[-1, -\rho)$). Similarly, we say that S is on the *opposite side* of the vehicle if the vehicle is located at ρ (resp. $-\rho$) and $S \subset [-1, -\rho]$ (resp. $[\rho, 1]$).

For $i \geq 1$, let S^i_{opp} and S^i_{same} be the intruders in an input instance I that arrive in the ith time interval that are on the opposite side and same side of the vehicle, respectively. Let $|S|$ denote the cardinality of S.

CAP operates as follows in the steady state, i.e., after time instant z. At any time instant $2j\rho+z$ for $j \geq 0$, the vehicle is stationed at either $-\rho$ or ρ. Without loss of generality, we assume that the vehicle is stationed at ρ. First, we observe that the intruders in S^{j+1}_{same} are located between ρ and $\rho+2\rho v$, the intruders in S^{j+2}_{same} are located between $\rho+2\rho v$ and $\rho+4\rho v$, and the intruders in S^{j+3}_{same} are located between $\rho+4\rho v$ and $\rho+6\rho v$ (Fig. 1). Further, because $z \geq 6\rho$, this means all the intruders in S^{j+3}_{same} have arrived by time $2j\rho+z$. Similar conclusions can be drawn for the intruders in S^{j+1}_{opp}, S^{j+2}_{opp}, and S^{j+3}_{opp}. If $|S^{j+2}_{\text{opp}}| > |S^{j+1}_{\text{same}}| + |S^{j+2}_{\text{same}}| + |S^{j+3}_{\text{same}}|$, then the vehicle moves to $-\rho$ arriving at time $2(j+1)\rho+z$ which is just in time to capture all the intruders in S^{j+2}_{opp}. If not, then the vehicle stays at ρ and captures all the intruders in S^{j+1}_{same} and reevaluates at time $2(j+1)\rho+z$. The key observation is that the vehicle moves from ρ to $-\rho$ only when it sees sufficient benefit in terms of the number of intruders in S^{j+2}_{opp} to sacrifice all the intruders in S^{j+1}_{same}, S^{j+2}_{same} and S^{j+3}_{same}.

For the initial case, the vehicle stays at the origin until

Algorithm 2: Capture with Patience Algorithm

Vehicle stays at origin from time 0 (time first intruder arrives) to time 2ρ, same side is right of origin
At time 2ρ, **if** $|S^1_{\text{opp}}| > |S^1_{\text{same}}|$ **then**
    Move to $x_2 = -\rho$, same side is left of origin
else
    Move to $x_2 = \rho$, same side unchanged
end
Wait until time z
for *each time instant* $z + 2\rho j, j \geq 0$ **do**
    if $|S^{j+2}_{\text{opp}}| > |S^{j+1}_{\text{same}}| + |S^{j+2}_{\text{same}}| + |S^{j+3}_{\text{same}}|$ **then**
        Move to $x_{j+1} = -x_j$, same side changes
    else
        Stay at x_j and capture interval S^{j+1}_{same}.
    end
end

time 2ρ. At time 2ρ, if $|S^1_{\text{opp}}| > |S^1_{\text{same}}|$ (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 vehicle will move to $-\rho$. Otherwise, the vehicle moves to ρ. In either case, the vehicle then stays at either $-\rho$ or ρ until time z.

Lemma 5 *CAP never moves the vehicle from ρ to $-\rho$ and then back to ρ (or vice versa) without capturing at least one interval of intruders.*

Proof: This holds as in order to move from ρ to $-\rho$ at time $2j\rho + z$, it must be true that $|S^{j+2}_{\text{opp}}| > |S^{j+1}_{\text{same}}| + |S^{j+2}_{\text{same}}| + |S^{j+3}_{\text{same}}|$. This implies $|S^{j+2}_{\text{opp}}| > |S^{j+3}_{\text{same}}|$. In order to move directly to ρ without capturing the intruders in the interval S^{j+2}_{opp}, we would need $|S^{j+3}_{\text{same}}| > |S^{j+2}_{\text{opp}}| + |S^{j+3}_{\text{opp}}| + |S^{j+4}_{\text{opp}}|$, but this cannot hold by the above observation. ■

Corollary IV.3 *For any $i \geq 1$, CAP will capture intruders from one of S^i_{same} or S^{i+1}_{opp}.*

Proof: Omitted for brevity. ■

Theorem IV.4 *CAP is 4-competitive for any environment $\mathcal{E}$ with $v \leq (1-\rho)/6\rho$.*

Proof: [Outline] We show that for any input instance I, CAP captures at least $1/4$th of all intruders in I using an accounting analysis where we "charge" lost intruders to captured intruders, or equivalently, captured intruders "pay" for the lost intruders. CAP only switches sides when the payoff on the other side is large enough to pay for the lost intruders on the original side. ■

We now prove some lower bounds on CAP's competitive ratio including showing that the bound is tight for some parameter settings.

Lemma 6 *CAP is no better than 3-competitive for $v \leq \frac{1-\rho}{6\rho}$ and is no better than 4-competitive for $v \leq \min\{\frac{1}{3}, \frac{1-\rho}{6\rho}\}$.*

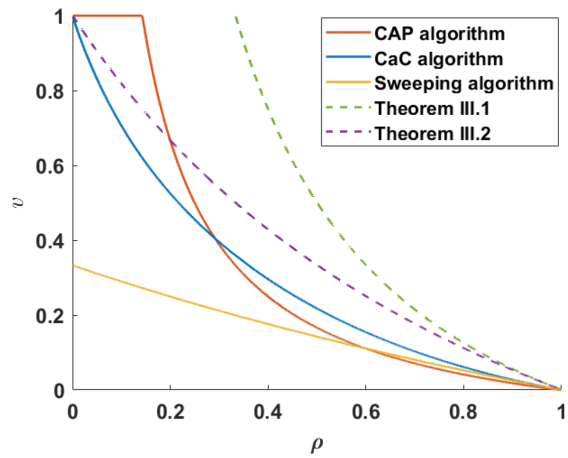

Fig. 2. Parameter regimes for our algorithms (solid lines, extend to the left) and lower bounds (dashed lines, extend to the right).

Proof: [Outline] We consider an input instance I consisting of two streams of intruders, one stream each arriving at endpoints 1 and -1. At location 1, one intruder arrives at time instant $6i\rho$, $0 \leq i \leq K$ for some large K. At location -1, one intruder arrives at time instant $3\rho + \frac{\rho}{v} + i2\rho$, $0 \leq i \leq K$. CAP stays at ρ and captures the less dense stream of intruders. Another algorithm stays at $-\rho$ and captures the dense stream of intruders regardless of parameter regime to prove the 3-competitive lower bound. For $v \leq \min\{\frac{1}{3}, \frac{1-\rho}{6\rho}\}$, we give an algorithm that captures all intruders yielding the 4-competitive lower bound. ■

V. Summary and Numerical Performance

A. Summary of the results

Figure 2 shows a v-ρ plot summarizing our results. For $\rho > 0.3$, as the curve defined by the conditions in Theorem IV.2 for the CaC algorithm is above the curve defined by the conditions in Theorem IV.4 for the CAP algorithm, one should implement the CaC algorithm rather than the CAP algorithm. This is because the CaC algorithm guarantees to capture at least half of the total number of intruders, whereas, the CAP algorithm captures only $\frac{1}{4}^{th}$ of the total number of intruders. For any $\rho \leq 0.3$, there exist values of v such that one might choose any of the three algorithms. The curve defined by the conditions in Theorem IV.2 for the CaC algorithm is completely below the curve defined by the condition in Theorem III.2. This suggests that for the values of v and ρ that lie above the curve defined by the conditions for the CaC algorithm, either there may still exist an algorithm which is 2-competitive or it may be possible to tighten the analysis of Theorem III.2 and Lemma 3.

B. Numerical Performance

We now analyze the average case performance, as opposed to the worst case performance, of our algorithms numerically. Of specific interest is the case when the intruders are generated stochastically as per a spatio-temporal process [10].

We performed numerical analysis of our algorithms using the following procedure. A Poisson process with rate λ

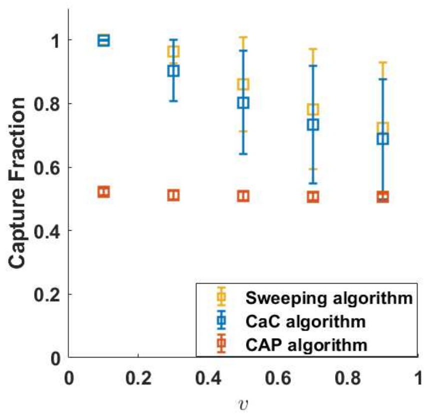

Fig. 3. Simulation result for $\lambda = 5$ and $\rho = 0.2$. The error bars indicate ± 1 standard deviation.

was used to model the arrival process of the intruders. The intruders arrive with equal probability on the endpoints. We simulated 50 runs per algorithm and present the mean and standard deviation of the *capture fraction* obtained by each algorithm for various values of v keeping ρ and λ fixed to 0.2 and 5 respectively. The *capture fraction* is defined as the ratio of the total number of intruders captured to the total number of intruders arrived in the environment [21]. The value of λ was kept high because for low arrival rate ($\lambda \to 0$), the number of intruders that arrive in the environment were very few and the capture fraction obtained was misleadingly high.

Figure 3 shows the simulation result for each of the algorithms. For values of v and ρ that lie above the blue curve in Figure 2, the CaC algorithm captured, on average, more than half of the intruders that arrived. Furthermore, the capture fraction of CAP algorithm was approximately 0.5 on average for all values of v. This is because the intruders being uniformly distributed, the vehicle can just capture intruders on one side and still ensure at least half of the total intruders are captured. Moreover, as v increases the capture fraction of Sweep algorithm approaches that of CaC algorithm. This is because as v increases, the sizes of the sets S_{same} and S_{opp} increase and eventually they cover the entire environment, thereby, converging to the Sweep algorithm.

VI. Conclusion and Future Work

This paper addressed a problem in which a single vehicle is tasked to defend a line segment perimeter from intruders. The key novelty of this work is an integration of concepts and techniques from competitive analysis of online algorithms with pursuit of multiple mobile intruders. We designed and analyzed three algorithms, i.e., Sweeping, Compare and Capture, and Capture with Patience algorithms, and demonstrated that they are 1, 2 and 4-competitive, respectively. We also derived fundamental limits on c-competitiveness for any constant c.

We plan to extend this work for the case when the intruders need to move outward or can actively evade the vehicle in order to reach the perimeter. Cooperative multi-vehicle defense in higher dimensional environments that can yield lower competitive ratios is another future direction.

References

[1] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of ACM*, vol. 28, no. 2, pp. 202–208, 1985.

[2] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corp., 1999.

[3] 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.

[4] E. Garcia, A. Von Moll, D. W. Casbeer, and M. Pachter, "Strategies for defending a coastline against multiple attackers," in *2019 IEEE 58th Conf. on Decision and Control (CDC)*. IEEE, 2019, pp. 7319–7324.

[5] 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.

[6] 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.

[7] 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.

[8] D. Shishika and V. Kumar, "Perimeter-defense game on arbitrary convex shapes," *arXiv preprint arXiv:1909.03989*, 2019, online available at:https:https://arxiv.org/abs/1909.03989.

[9] H. N. Psaraftis, "Dynamic vehicle routing problems," *Vehicle routing: Methods and studies*, vol. 16, pp. 223–248, 1988.

[10] D. J. Bertsimas and G. Van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[11] S. L. Smith, M. Pavone, F. Bullo, and E. Frazzoli, "Dynamic vehicle routing with priority classes of stochastic demands," *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3224–3245, 2010.

[12] S. D. Bopardikar and V. Srivastava, "Dynamic vehicle routing in presence of random recalls," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 37–42, 2019.

[13] 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.

[14] E. Angelelli, M. Grazia Speranza, and M. W. Savelsbergh, "Competitive analysis for dynamic multiperiod uncapacitated routing problems," *Networks: An International Journal*, vol. 49, no. 4, pp. 308–317, 2007.

[15] M. Blom, S. O. Krumke, W. E. de Paepe, and L. Stougie, "The online TSP against fair adversaries," *INFORMS Journal on Computing*, vol. 13, no. 2, pp. 138–148, 2001.

[16] S. Henn, "Algorithms for on-line order batching in an order picking warehouse," *Computers & Operations Research*, vol. 39, no. 11, pp. 2549–2563, 2012.

[17] A. Stieber, A. Fügenschuh, M. Epp, M. Knapp, and H. Rothe, "The multiple traveling salesmen problem with moving targets," *Optimization Letters*, vol. 9, no. 8, pp. 1569–1583, 2015.

[18] M. Hassoun, S. Shoval, E. Simchon, and L. Yedidsion, "The single line moving target traveling salesman problem with release times," *Annals of Operations Research*, pp. 1–10, 2019.

[19] C. S. Helvig, G. Robins, and A. Zelikovsky, "Moving-target TSP and related problems," in *European Symposium on Algorithms*. Springer, 1998, pp. 453–464.

[20] S. Bajaj and S. D. Bopardikar, "Dynamic boundary guarding against radially incoming targets," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 4804–4809.

[21] S. L. Smith, S. D. Bopardikar, and F. 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*, 2009, pp. 8543–8548.

[22] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 2005.

[23] S. O. Krumke, W. E. De Paepe, D. Poensgen, and L. Stougie, "News from the online traveling repairman," *Theoretical Computer Science*, vol. 295, no. 1-3, pp. 279–294, 2003.

[24] S. Bajaj, E. Torng, and S. D. Bopardikar, "Competitive perimeter defense on a line," *arXiv preprint arXiv:2103.11787*, 2021, online available at: https://arxiv.org/abs/2103.11787.