The Role of Heterogeneity in Autonomous Perimeter Defense Problems*

Aviv Adler[1[0000−0003−3698−7639]], Oscar Mickelin[2[0000−0003−0167−1992]], Ragesh K. Ramachandran[3[0000−0003−4396−0127]], Gaurav S. Sukhatme[3,5[0000−0003−2408−474X]], and Sertac Karaman[3[0000−0002−2225−7275]]

[1] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA adlera@mit.edu

[2] Program in Applied and Computational Mathematics, Princeton University, Princeton NJ 08544, USA hm6655@princeton.edu

[3] Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA. {rageshku,gaurav}@usc.edu

[4] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 02139 sertac@mit.edu

[5] G.S. Sukhatme holds concurrent appointments as a Professor at USC and as an Amazon Scholar. This paper describes work performed at USC and is not associated with Amazon.

Abstract. When is heterogeneity in the composition of an autonomous robotic team beneficial and when is it detrimental? We investigate and answer this question in the context of a minimally viable model that examines the role of heterogeneous speeds in perimeter defense problems, where defenders share a total allocated speed budget. We consider two distinct problem settings and develop strategies based on dynamic programming and on local interaction rules. We present a theoretical analysis of both approaches and our results are extensively validated using simulations. Interestingly, our results demonstrate that the viability of heterogeneous teams depends on the amount of information available to the defenders. Moreover, our results suggest a universality property: across a wide range of problem parameters the optimal ratio of the speeds of the defenders remains nearly constant.

Keywords: Perimeter defense · Heterogeneous multi-robot team · Dynamic Programming.

1 Introduction

An increasingly important task, where a robotic system can be employed, is in defending an area against external agents, which pose varying levels of threat. Examples include defending airports against intruding and flight-grounding drones

* Supported by the Army Research Laboratory as part of the Distributed and Collaborative Intelligent Systems and Technology (DCIST) Collaborative Research Alliance (CRA).

[7], defending wildlife habitats against trespassing poachers [2], extinguishing and preventing the spread of devastating wildfires caused by human or natural activity [9], as well as military applications [14].

In general, solutions to perimeter defense problems allude to finding strategies for a set of agents restricted to the perimeter of an area, entrusted with defending the area from intruders which are trying to breach the perimeter of the area [17].

Compared to a homogeneous team of robots, a team of robots with varying capabilities (heterogeneous team) comes with its unique set of advantages and challenges. Equipping different agents with different capabilities can lead to synergy effects where the heterogeneous system outperforms the alternative homogeneous system composed of identical agents. As a result, in the last decade, there has been significant interest in the robotics community to define, explore, and quantify heterogeneity in different robot applications [20, 15, 12, 8, 13, 11].

This paper investigates the impact of heterogeneity in multi-robot teams for the perimeter defense problem. We propose two optimal strategies, valid under different assumptions. The first strategy is based on dynamic programming (DP) [3]. It is optimal when the defenders are able to predict the location of the incoming attacks, but suffers from the curse of dimensionality and therefore relatively high associated computational costs. The second strategy is based on local interaction rules, and is optimal when the defenders have no information about the incoming attacks. This strategy can be efficiently computed in an online fashion, but does not implement any prior knowledge of the attack locations.

We prove the optimality of both strategies and analyze their time complexities. The algorithms are extensively validated on simulations. Our numerical experiments are two-dimensional, but the majority of the theoretical results remain valid for any dimension. This includes three-dimensional perimeters in applications involving drones, and higher-dimensional perimeters arising as constraint sets in a state space of arbitrary dimension.

Our results show that heterogeneity is beneficial in the case where the defenders have access to information about the incoming attacks, and is detrimental when the defenders have no information about the attacks. Moreover, we show the universality property that the optimal ratio of the speeds of the defenders remains nearly constant for a two defender case setting.

Related work: Perimeter defense problems are a variant of pursuit-evasion problems which have been studied extensively in literature. The seminal work of Issacs delineates differential-game approaches to arrive at equilibrium strategies for one pursuer one evader games [5]. There has been considerable effort by researchers from various communities for solving various variants of pursuit-evasion games involving multiple pursuers and evaders [21, 22, 4]. These papers contain works that view pursuit-evasion games either from the pursuers' side, from the evaders' side, or both. The curse of dimensionality poses a considerable challenge in solving problems involving multiple pursuers and evaders. The perimeter defense problem presented in this paper is a variant of the *target guarding problem* first introduced by Isaacs [5]. In the target guarding problem setting an agent is tasked with guarding an region of interest against an adver-

Table 1: Notations

Symbol	Description
$\mathcal{X}$	Perimeter
m	Number of defenders
n	Number of attacks
x_i	Location of defender i
v_i	Speed of defender i, ordered decreasingly
z_j	Location of attack j
t_j	Time of attack j, ordered increasingly
h	Defender horizon
$\text{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$	Minimum number of attacks the defenders can let through

sarial agent. Investigations on perimeter defense problems are in their nascent stage. The review paper by Shishika and Kumar [17] delineates the recent works done on multi-robot perimeter defense problems [16, 6, 19, 18]. Unlike the problems considered in these works, we consider a class of perimeter defense problems in which the number of attackers is much larger than the number of defenders.

The remainder of the paper is organized as follows. Section 2 contains our notation together with the problem statement. Sections 3 and 4 detail our theoretical results in the infinite and unit-time horizon cases respectively. Section 5 concludes with simulation results.

2 Problem statement

In this paper, bold letters are used to represent vectors and non-bold letters to represent scalars. Calligraphic letters are used to represent sets, and $|\mathcal{S}|$ denotes the cardinality of a set $\mathcal{S}$.

For any positive integer $n \in \mathbb{Z}^+$, $[n]$ denotes the set $\{1, 2, \cdots, n\}$. For a domain $\mathcal{X}$ with $x_1, x_2 \in \mathcal{X}$, $\text{dist}(x_1, x_2)$ denotes the length of the shortest path between x_1 and x_2 contained inside $\mathcal{X}$. As an example, in the case when $\mathcal{X}$ denotes a circle of radius $\frac{1}{2\pi}$

$$\text{dist}(x_1, x_2) = \frac{1}{2\pi}\min\left(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|\right), \tag{1}$$

where θ_1, θ_2 are the polar angles of x_1 and x_2, respectively.

2.1 Perimeter defense against point attacks

For ease of reference, the notation of this section is summarized in Table 1. Our problem is perimeter defense against point attacks with mobile defenders of varying speeds. Specifically, we have a perimeter $\mathcal{X}$ in d-dimensional space, with a distance metric dist, defended by m mobile defenders with speeds $v_1, \ldots, v_m$, so that defender i at $x \in \mathcal{X}$ at time t can make it to x' at time t' iff

$$\text{dist}(x, x') \le (t' - t)v_i \tag{2}$$

Without loss of generality we order the defenders from fastest to slowest, i.e. $v_1 \geq \cdots \geq v_m$, and we denote the *speed vector* as $\boldsymbol{v} = (v_1, \ldots, v_m)$. Then n attacks $(z_j, t_j) \in \mathcal{X} \times \mathbb{R}_{\geq 0}$, where z_j is the location on $\mathcal{X}$ at which it happens, and t_j is the time; WLOG we order these by time, i.e. $t_1 \leq \cdots \leq t_n$. Because attacks happen at fixed locations and times, they cannot react to the positions of the defenders.

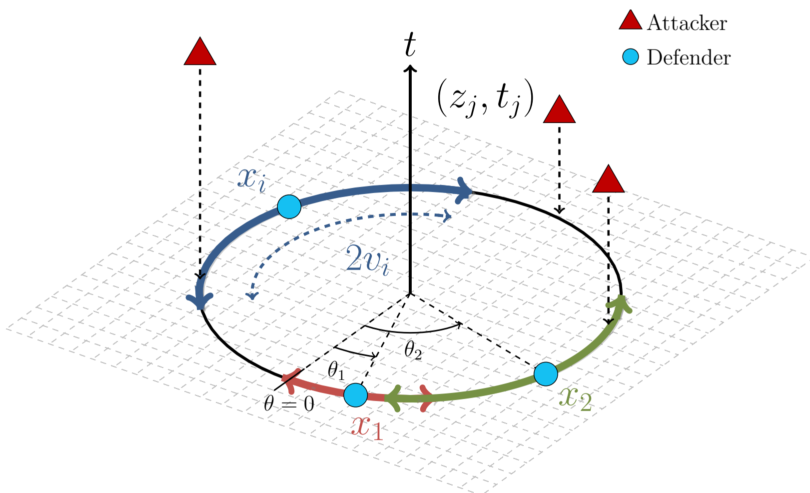

Fig. 1: Three defenders facing three attacks, with the unit-time reachable sets for each defender shown. Note that the third dimension is time; if the attack represents a physical object it is approaching from somewhere outside the circle, but we are only concerned with where and when it will hit the perimeter. In this example the defenders are not allowed to leave the perimeter, so the size of the reachable set scales linearly with speed (until it covers the whole perimeter).

An attack (z_j, t_j) is *thwarted* if and only if a defender is present, i.e. there is some defender i at z_j at time t_j; otherwise, we say that the attack *breaches* the perimeter. The goal is to design a policy for the defenders that minimizes the number of attacks that breach the defenses, and to study the effectiveness of different defender speed combinations against attacks.

Additionally, the team of defenders has a *horizon* h under which they can see attacks: specifically, at time t, any attack (z_j, t_j) is known to the defenders if and only if $t_j \leq t + h$. We will study in particular the unit horizon $h = 1$ and the infinite horizon $h = \infty$ (all attacks are visible from the start).

Finally, the defenders are allowed to start at $t = 0$ at any combination of locations in $\mathcal{X}$; they are even allowed to choose their starting locations based on the attack sequence (up to horizon h).

Given a speed vector $\boldsymbol{v}$ and sequence of attacks $\{(z_j, t_j)\}_{j=1}^n$, we define $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ as the minimum number of attacks from $\{(z_j, t_j)\}_{j=1}^n$ that defenders of speed $\boldsymbol{v}$ can let through (with all attacks known). In some cases we will be dealing with $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ for one sequence of attacks $\{(z_j, t_j)\}_{j=1}^n$ over many defender speed vectors $\boldsymbol{v}$; in that case we write $\mathrm{Opt}(\boldsymbol{v})$ for convenience.

2.2 Different settings

Within the above problem description, there are several different variations, mostly to do with how the attacks are generated and the length of the horizon h. We roughly divide attack sequences into two settings:

1. Any sequence of attacks (z_j, t_j) is legitimate.
2. Attacks must come at unit time intervals, i.e. $t_j = j$ for all $j \in [n]$.

Note that in setting 2 we do not lose any generality by having the attacks happen at unit time intervals, since we can rescale the time units (and adjust the speeds of the defenders accordingly). Since the index j is superfluous in setting 2 we refer to the sequence of attacks as $z_1, z_2, \ldots, z_n$, indexed by t.

In setting 1, we study the case where all attacks are known to the defenders at the start; our primary problems are (i) find an algorithm for the defenders' movements that minimizes the number of breaches, and (ii) study the behavior of optimal defense against uniformly-random attacks (in both location and time) for different combinations of defenders. Since setting 1 is more general, the algorithms will also apply to setting 2.

In setting 2, we study the case where the attacks are (i) generated uniformly at random in location (time is fixed) and (ii) generated by an adversary which wants to guarantee a breach with as few attacks as possible. We also consider both the case where all the attacks are known to the defenders at the start ($h = \infty$) and the case where attack t only becomes known at time $t - 1$ ($h = 1$).

Remark 1. Here we deal with the case where the number of defenders is fixed, and the question is how fast to make each defender (and in particular whether to make them all the same speed or not). The alternative case of varying the number of defenders is investigated in the arxiv version of this work [1], especially in regards to the tradeoff between fewer and faster defenders versus more and slower ones.

3 Infinite Horizon Theoretical Results

3.1 Dynamic programming with infinite horizon

We now give an algorithm which, given defender speeds $\boldsymbol{v} = (v_1, \ldots, v_m)$ and attacks $\{(z_j, t_j)\}_{j=1}^n$ returns two things: (i) $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ (the minimum number of attacks that can be let through); and (ii) the list (of lists) $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m)$, where ℓ_i is the (sub)sequence of attacks which defender i should thwart. We refer to $\boldsymbol{\ell}$ as a *defense plan*.

Recall that by default the attacks are sorted in order of arrival time (or the user should sort them before applying the algorithm).

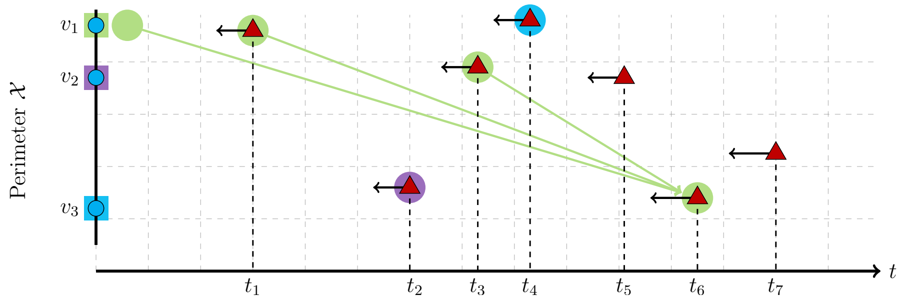

Fig. 2: Computing $f(6, 2, 4)$ (defender 1 has to thwart attack 6, etc.) recursively; each defender is allowed to thwart attacks prior to these, but not afterwards. Since 6 is the maximum value, we consider the last attack that defender 1 can handle before 6: based on its speed, it can be 0 (defend nothing before 6), 1, or 3. Thus $f(6, 2, 4) = \min(f(0, 2, 4), f(1, 2, 4), f(3, 2, 4)) - 1$.

The pseudocode is given in Alg. 1, in which we use the following notation: $\boldsymbol{j} = (j_1, \ldots, j_m) \in \{0, 1, \ldots, n\}^m$ denotes a vector of attacks assigned to each defender (with $j_i = 0$ indicating no attack assigned to defender i, and we allow the j_i's to be non-distinct even though it is redundant);

$$\boldsymbol{j}_{-i}(j') = (j_1, \ldots, j_{i-1}, j', j_{i+1}, \ldots, j_m) \tag{3}$$

i.e. $\boldsymbol{j}$ with the ith entry replaced by j';

$$\mathbf{1}_i(j', j'') := \begin{cases} 1 & \mathrm{dist}(z_{j'}, z_{j''}) \leq (t_{j''} - t_{j'})v_i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

is the indicator that defender j is capable of thwarting attack j'' after thwarting j' (and $\mathbf{1}_i(0, j'') = 1$ since defenders can start anywhere); $[\cdot] + [\cdot]$ denotes concatenation (of lists); and arg min (arg max) denote the sets of values minimizing (maximizing) the arguments. The for-loop in Alg. 1 iterates in lexicographic order, skipping $f(0, \ldots, 0)$ (which is already known) so the recursion can work.

The proof of the following result is in the arxiv version of this work [1]:

Theorem 1. *Alg. 1 outputs the correct value of* $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ *and* $\boldsymbol{\ell}$.

Alg. 1 depends on the function $f(\boldsymbol{j}) : \{0, 1, \ldots, n\}^m \to \mathbb{N}$, which denotes the following: suppose that defender i (with speed v_i) is required to thwart attack j_i *and then no others after that* (but defender i can thwart attacks arriving before t_{j_i}, and if $j_i = 0$, then defender i is not allowed to thwart any attack); $f(\boldsymbol{j})$ is the minimum number of defenders that can be let through under these constraints. Then the following hold:

- $f(0, \ldots, 0) = n$ (the base case from which we recursively compute f);
- $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n) = \min_{\boldsymbol{j}} f(\boldsymbol{j})$ (this allows us to extract the correct value by keeping track of this minimum).

It then recursively computes $f(\boldsymbol{j})$ for all $\boldsymbol{j} \in \{0,1,\dots,n\}^m$; see Figure 2 for an example and the arxiv version [1] for the details.

Algorithm 1: Dynamic programming for infinite horizon defenders.

Data: Attacks $\{(z_j,t_j)\}_{j=1}^n$; defender speeds $\boldsymbol{v} = (v_1,\dots,v_m)$
Result: $\mathrm{Opt}(\boldsymbol{v}, \{(z_j,t_j)\}_{j=1}^n)$
1 $f(0,\dots,0) = n,\ \ c = n,\ \ \boldsymbol{j}^{\min} = (0,\dots,0)$; `/* initialization */`
2 **for** $\boldsymbol{j} \in \{0,1,\dots,n\}^m$ **do** `/* compute f */`
3 $\quad \mathcal{S} = \arg\max_i \{j_i\}$; `/*` $\mathcal{S}$ `is the set of those` i `that minimize` j_i `*/`
4 $\quad$ Choose any $i^*(\boldsymbol{j}) \in \mathcal{S}$;
5 $\quad$ Choose any $j^*(\boldsymbol{j}) \in \arg\min_{j'} \left\{ f(\boldsymbol{j}_{-i^*(\boldsymbol{j})}(j')) : j' < j_{i^*(\boldsymbol{j})} \text{ and } \mathbf{1}_{i^*}(j', j_{i^*(\boldsymbol{j})}) \right\}$
6 $\quad$ **if** $|\mathcal{S}| = 1$ **then**
7 $\quad\quad f(\boldsymbol{j}) = f(\boldsymbol{j}_{-i^*(\boldsymbol{j})}(j^*(\boldsymbol{j}))) - 1$
8 $\quad$ **else**
9 $\quad\quad f(\boldsymbol{j}) = f(\boldsymbol{j}_{-i^*(\boldsymbol{j})}(j^*(\boldsymbol{j})))$
10 $\quad$ **end**
11 $\quad$ **if** $f(\boldsymbol{j}) < c$ **then**
12 $\quad\quad c = f(\boldsymbol{j}),\ \ \boldsymbol{j}^{\min} = \boldsymbol{j}$
13 $\quad$ **end**
14 **end**
15 $\boldsymbol{\ell} = (\ell_1,\dots,\ell_m) = ([j_1^{\min}],\dots,[j_m^{\min}])$; `/* initialize defender lists */`
16 $\boldsymbol{j}^{\mathrm{curr}} = \boldsymbol{j}^{\min}$
17 **while** $\boldsymbol{j}^{curr} \neq (0,\dots,0)$ **do** `/* reconstruct defender lists */`
18 $\quad$ **if** $j^*(\boldsymbol{j}) \neq 0$ **then**
19 $\quad\quad \ell_{i^*(\boldsymbol{j})} = [j^*(\boldsymbol{j})] + \ell_{i^*(\boldsymbol{j})}$; `/* add` $j^*(\boldsymbol{j})$ `to front of list */`
20 $\quad$ **end**
21 $\quad j^{\mathrm{curr}}_{i^*(\boldsymbol{j})} = j^*(\boldsymbol{j})$
22 **end**
23 **return** $\mathrm{Opt}(\boldsymbol{v}, \{(z_j,t_j)\}_{j=1}^n) = c,\ \ \boldsymbol{\ell}$

Remark 2. Alg. 1 relies on the subtle point that $i^*(\boldsymbol{j}) \in \arg\max_i j_i$ because if not, then we do not know whether to subtract 1 when we do the update; by setting $i^*(\boldsymbol{j}) \in \arg\max_i j_i$, we remove the question of whether a defender i' assigned to a later $j_{i'}$ can also thwart attack j_{i^*}.

Remark 3. Alg. 1 assumes that the defenders can start at whatever locations they want, but can be modified for fixed defender starting locations (or a set of possible starting locations) by redefining $\mathbf{1}_i(0,j)$ to indicate whether they can reach attack j from their starting locations. It can also be modified for the important case where attacks cause varying amounts of damage, with attack j doing w_j damage (should it not be intercepted); see for instance the Iron Dome missile defense system, which prioritizes attacks based on potential damage estimates [14]. To make this modification, replace -1 with $-w_{j_{i^*(\boldsymbol{j})}}$ in line 7 and $f(0,\dots,0) = c = n$ with $f(0,\dots,0) = c = \sum_j w_j$.

Given m defenders and n attackers, the number of computations needed to run Alg. 1 is on the order of $(n+1)^{m+1}$ (we need to run through $(n+1)^m$ values of $\boldsymbol{j}$, and each update takes up to n time for the comparisons).

3.2 Monotonicity-based computational acceleration

In order to investigate team heterogeneity, we compute $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ for all $\boldsymbol{v}$ whose elements v_i are at g evenly-spaced locations in a range $(v_{\min}, v_{\max}]$.[6] We refer to g as the number of *grains*. If we were to run Alg. 1 for all combinations $\boldsymbol{v}$ of speeds, the complexity becomes $O((n+1)^{m+1}g^m)$, which gets extremely large very quickly.

However, as each attack sequence is evaluated on all $\boldsymbol{v}$, we can take advantage of the monotonicity of Opt over $\boldsymbol{v}$ to reduce the amount of computation needed.

In particular, for any sequence $\{(z_j, t_j)\}_{j=1}^n$,

$$\boldsymbol{v} \leq \boldsymbol{v}' \implies \mathrm{Opt}(\boldsymbol{v}) \geq \mathrm{Opt}(\boldsymbol{v}') \tag{5}$$

since faster defenders can always emulate slower ones and thus achieve (at least) as good a result on any attack sequence. This means that

$$\mathrm{Opt}(\boldsymbol{v}) = \mathrm{Opt}(\boldsymbol{v}') = k \text{ for some } \boldsymbol{v} \leq \boldsymbol{v}' \tag{6}$$

$$\implies \mathrm{Opt}(\boldsymbol{v}'') = k \text{ for all } \boldsymbol{v} \leq \boldsymbol{v}'' \leq \boldsymbol{v}' \,. \tag{7}$$

Thus we know $\mathrm{Opt}(\boldsymbol{v}'') = k$ for a range of $\boldsymbol{v}''$, without having to run Alg. 1. Taking the set of values $\boldsymbol{v} \in (v_{\min}, v_{\max}]^m$ (of given grains), for any $\{(z_j, t_j)\}_{j=1}^n$ we can evaluate $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ in a strategic order to minimize the number of times we need to run Alg. 1. This is discussed in greater detail in the arxiv version of this work [1].

4 Unit Horizon Theoretical Results

This section considers defenders with a unit horizon of incoming attacks. The general setup is

- We consider two defenders with speeds $v_1 \geq v_2$.
- We consider a perimeter $\mathcal{X}$ homeomorphic to $\mathcal{S}^1$ (a circle[7]), with distances determined by arc length and total length normalized to 1; we represent $\mathcal{X} = [-1/2, 1/2]$ (but $-1/2$ and $1/2$ are the same point). To denote this situation, we define the distance function

$$\mathrm{dist}(y_1, y_2) = \min\left\{|y_1 - y_2|, 1 - |y_1 - y_2|\right\} \tag{8}$$

  (a rescaled version of (1)). The maximum possible value of $\mathrm{dist}(y_1, y_2)$ is $1/2$, and we assume they start at maximum distance from each other, i.e., at antipodal points.
- The n attackers are generated according to Setting 2 from Section 2.2: attacker t appears at time t, uniformly (and independently) over $\mathcal{X}$.

[6] For instance, if $g = 5$ and $(v_{\min}, v_{\max}] = (0, 1]$, we measure $\boldsymbol{v}$ where $v_i \in \{0.2, 0.4, 0.6, 0.8, 1\}$ for all i.

[7] We consider this case because it has a number of nice symmetries, and because perimeters enclosing a simply-connected 2D area are homeomorphic to $\mathcal{S}^1$.

– The defenders have a unit horizon in time: at any given time they only see the next incoming attack, though they also know n and the current time t.

Therefore the defenders' policy can be thought of as a sequence of decisions taken at unit time intervals (i.e. when the next attack is revealed), which is naturally formulated as a Markov Decision Process (MDP) [10] with n steps, with the reward being the number of thwarted attacks.

To simplify the MDP we can remove one state variable since, by symmetry, we can rotate $\mathcal{X}$ (or relabel it) so that at the beginning of any time step, defender 1 is at location 0. We can also reflect it so that defender 2 is on the positive half. Thus the state at time t (just before the location of the next attack is revealed) can be denoted by a single parameter $a(t)$, indicating the distance between the two defenders. Then the next attack's location $x(t+1)$ is revealed, in the coordinate system relative to the defenders' positions.

4.1 Policy and Reward

A *unit-horizon policy* is a function $f : [0, 1/2] \times [-1/2, 1/2] \to [0, 1/2]$. The inputs are $a(t)$, $x(t)$ and the number of remaining attacks, and the output is $f(a(t), x(t)) = a(t+1)$. As described above, $a(t+1)$ is the distance between the two defenders at time $t+1$. f must satisfy the condition

$$f(a(t), x(t)) \in [a(t) - v_2 - v_1, a(t) + v_2 + v_1] \tag{9}$$

The policy then produces a reward

$$r(t) := r(a(t), x(t), f(a(t), x(t))) \in \{0, 1\} \tag{10}$$

the reward, based on whether the given movement makes it possible for the attack to be thwarted ($r(t) = 1$ if so, $= 0$ if not). $r(t)$ is given as follows:

$$r(t) = \begin{cases} 1 & \text{if } \operatorname{dist}(0, x(t)) \le v_1 \text{ and } [\operatorname{dist}(x(t), a(t)) - v_2, \operatorname{dist}(x(t), a(t)) + v_2] \\ 1 & \text{if } \operatorname{dist}(a(t), x(t)) \le v_2 \text{ and } f(a(t), x(t)) \in [x(t) - v_1, x(t) + v_1] \\ 0 & \text{otherwise} \end{cases}$$

The reason for this is that by symmetry (of the perimeter and of the attacks), given the distance $a(t+1) = f(a(t), x(t))$ between the defenders at the start of the next step, the ability of the defenders to stop future attacks does not depend on their locations. Thus, if the defenders can stop the current attack and end at distance $a(t+1) = f(a(t), x(t))$ for the next step, this is always preferable to ending at the same distance *without* making the capture.

Hence $r(t) = 1$ under policy f if and only if this is possible, which can be split into two cases: (i) defender 1 makes the capture; (ii) defender 2 makes the capture. If either of these are feasible, $r(t) = 1$; if neither are, $r(t) = 0$.

Remark 4. If $\operatorname{dist}(a(t), x(t)) > v_2$ and $\operatorname{dist}(0, x(t)) > v_1$, this means that neither defender can reach the next attack and hence $r(t) = 0$ no matter what.

4.2 Optimal defender policy

Fix a defender policy f. For a given total number N of incoming attacks and an initial distance a between the two defenders, we define the expected reward $J(a;N)$ of the defenders as the expected total number of thwarted attacks, i.e.,

$$J(a;N) := \mathbb{E}_x \left[\sum_{t=0}^{N-1} r(t) \right] \text{ under policy } f, \tag{11}$$

where the expectation is over the attack locations $x(t)$. With this definition, we are interested in determining the policy f that leads to the highest expected reward. We show in our arxiv version [1] that for a wide range of values for v_1, v_2 and N, the optimal strategy should (i) always thwart the currently-known if possible. We next prove that the optimal policy subsequently should (ii) always maximize $a(t)$ subject to the first constraint. That is:

Proposition 1. *f^* maximizes $J(a;N)$ if (ii) $a(t{+}1)$ is maximized for all inputs, over all policies that satisfy (i) (i.e. capture when possible).*

We next show necessary and sufficient conditions for perfect defense, i.e. when no (fixed-time) attack sequence can force a breach.

Theorem 2 (The perfect defense theorem). *For any pair of defenders with speeds v_1, v_2 where $v_2 \leq v_1$, there exists a sequence of attacks that breaches if and only if $v_1 < 1/2$ and $v_1 + 3v_2 < 1$. Furthermore, if $v_1 \geq 1/2$ or $v_1 + 3v_2 \geq 1$, the defenders can defend indefinitely even with a one-step horizon. Furthermore, if any sequence of attacks guarantees a breach, there is a sequence of at most 6 attacks that does so.*

Both proofs are given in the arxiv version of this work [1].

5 Simulation Results

We conduct simulations for each of the settings from Section 2.2. Our experiments are run as follows:

1. Generate attacks $\{(z_j, t_j)\}_{j=1}^n$ randomly, either with fixed attack times $t_j = j$ or uniformly-random attack times in $[0, t_{\max}]$.
2. Compute $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^n)$ for $\boldsymbol{v} \in (v_{\min}, v_{\max}]^m$, at g intervals.
3. Repeat the above for T trials and average the resulting values for each $\boldsymbol{v}$.

We conduct all of our experiments on a circular perimeter of circumference 1, where the defenders are not permitted to leave the perimeter (so maximally distant points are at opposite ends and have distance 1/2). Comparison of the results sheds light on the conditions which favor heterogeneous defender teams and those which favor homogeneous teams and/or single super-defenders.

Table 2: Parameters of the experiments

Symbol	Description
m	Number of defenders ($m = 2$ unless specified otherwise)
n	Number of attacks
T	Number of trials
$t_{\max}$	Size of attack window (not needed for heterogeneous setting (ii))
$(v_{\min}, v_{\max}]$	Range of defender speeds (inclusive of $v_{\max}$ but not $v_{\min}$)
g	Number of speed values measured (grains) within $(v_{\min}, v_{\max}]$

The structure of the simulations means each combination of defender speeds is evaluated on the same set of attack sequences, which makes the comparison fairer, and allows us to significantly speed up the computation when evaluating $\mathrm{Opt}(\boldsymbol{v}, \{(z_j, t_j)\}_{j=1}^{n})$ for many values of $\boldsymbol{v}$ on a single attack sequence $\{(z_j, t_j)\}_{j=1}^{n}$, by exploiting the fact that Opt is a monotonically-decreasing step function in $\boldsymbol{v}$ (as described in Section 3.2).

The full list of parameters is given in Table 2.

5.1 Simulation Results

In Figure 3, we simulate sequences of $n = 25$ attacks of both settings, where the perimeter $\mathcal{X}$ is a unit circle of circumference 1 and $m = 2$ defenders; for uniformly random attack times we set $t_{\max} = 25$ to get the same density of attacks in both cases. This is analyzed over the speed range $(v_{\min}, v_{\max}] = (0, 0.6]$ with $g = 256$ grains. The left column shows results for uniformly-random attack times; the right column shows results for fixed attack times.

The results are given as surface plots, taking defender speeds v_1, v_2 and returning $\mathrm{Opt}(v_1, v_2)$ (ignoring the assumption in the analysis that $v_1 \geq v_2$, so the plots are symmetric about the line $v_1 = v_2$). We give:

- **Top row:** $\mathrm{Opt}(v_1, v_2)$ for a single sequence of attacks. This can be viewed as $T = 1$, and is meant to give a visualization of how adjusting the speeds of the defenders changes the ability to defend against a particular sequence. Since $\mathrm{Opt}(v_1, v_2)$ takes integer values, we have a monotonically-decreasing step function.
- **Middle and bottom rows:** $\mathrm{Opt}(v_1, v_2)$ when averaged over $T = 200$ randomly-generated attack sequences. Middle row gives the front view to show overall shape; bottom row gives the back view to show the ridge at $v_1 = v_2$. This ridge, which appears for both uniformly-random attack times and fixed attack times, shows that on average homogeneous defenders are less efficient (per combined speed) than heterogeneous defenders.

From this we can make a number of interesting observations:

- $\mathrm{Opt}(v_1, v_2)$ is generally larger for the uniformly random attack times, as attacks which are close together in time are much harder to defend. In particular, with fixed attack times $\mathrm{Opt}(v_1, v_2) = 0$ for sufficiently large defender speeds (one defender of speed $1/2$ is already sufficient to defend all attacks).

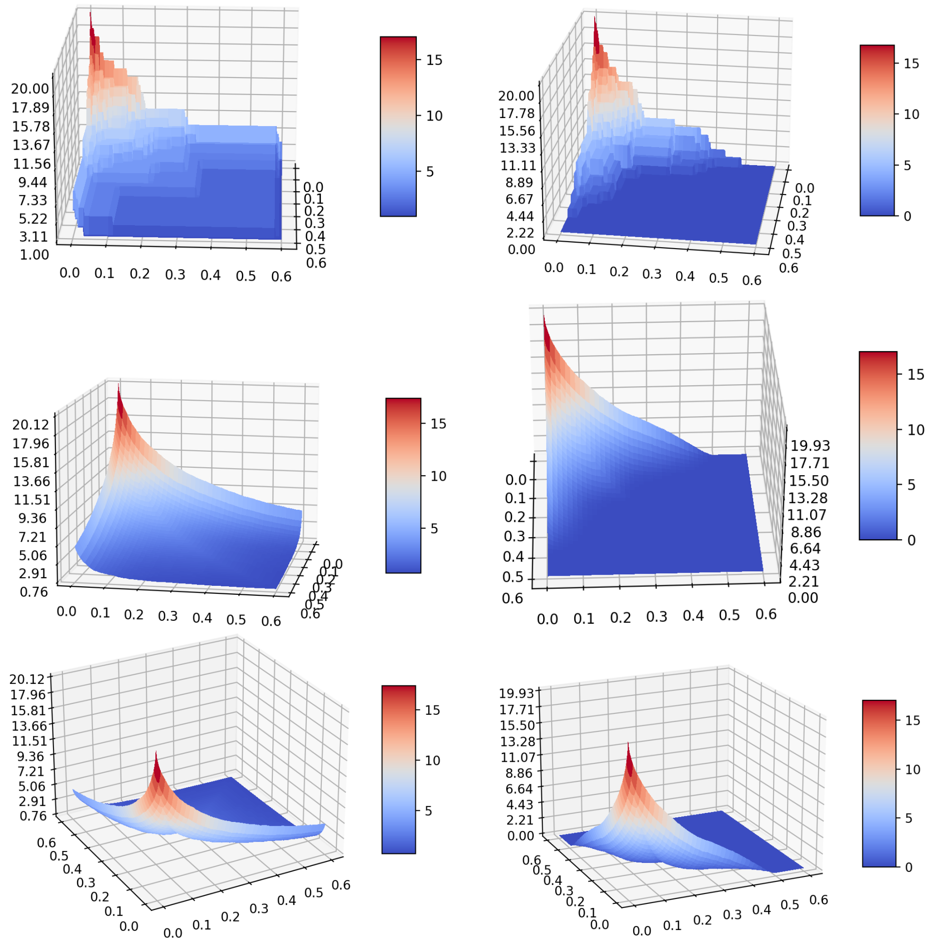

Fig. 3: 2 defenders evaluated at $g = 256$ grains for speeds $(v_{\min}, v_{\max}] = (0, 0.6]$ for 200 trials. *Top row:* 1 trial, front view. *Middle row:* 200 trials, front view. *Bottom row:* 200 trials, back view, showing the 'ridge' at the center line $v_1 = v_2$ (both attack types). **Left:** Uniformly-random attack times, $n = 25$, $t_{\max} = 25$; **Right:** Fixed attack times, $n = 25$.

- As mentioned, there is a ridge on $v_1 = v_2$ (the back view makes it clearly visible). This shows that on average, homogeneous defenders are less effective than well-designed heterogeneous ones.
- Under uniformly-random attack times, each 'half' (cutting at the $v_1 = v_2$ line) is empirically convex, while under fixed attack times, each 'half' is convex near the $v_1 = v_2$ ridge but becomes concave again near the edge of the plot (as seen in the back view) and as the defender speeds increase (as can be seen on the edge in both views).

We also consider the question: what is the optimal *mix* of defender speeds? To answer this, we need to consider what we want to hold constant, since obviously faster defenders are always better; an obvious starting point is to look at defenders of a fixed total speed, and consider what ratio of speeds performs the best. This also means that we are comparing defender teams whose reachable sets are of equal total size (ignoring overlaps), and (because we evaluate over a grid of possible values of $\boldsymbol{v}$) means we compare the values of $\text{Opt}(\boldsymbol{v})$ on a diagonal line.

In Figure 4, we show the best (empirical) mixture: for each value of $v_{tot} = v_1 + v_2$, the returned value is

$$\frac{v_2^*}{v_{tot}} \text{ where } v_2^* := \arg\min_{v_2 \leq v_{tot}/2} \text{Opt}((v_{tot} - v_2, v_2)) \tag{12}$$

That is, given a total speed of v_{tot}, what is the optimal fraction of the speed 'budget' to assign to the slower defender? A value of 0.5 signifies homogeneous defenders are best; a value of 0.0 signifies that a single super-defender is best; and a value in between signify some heterogeneous mix of defenders is best.

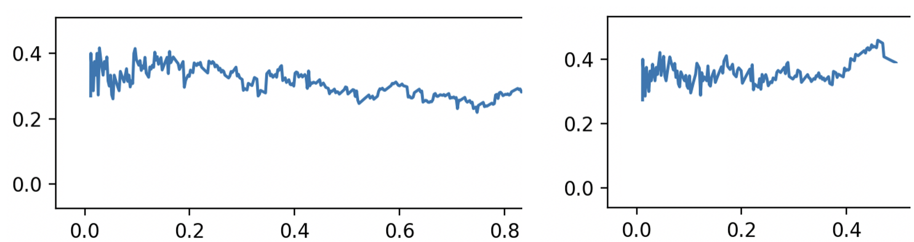

Fig. 4: Empirical optimal ratio v_2/v_{tot}, for various values of v_{tot}. *Left:* Uniform attack times. *Right:* Fixed attack times.

These are based on the same experiments as shown in Figure 3. Note that the fixed attack times graph ends at $v_{tot} = 0.5$; past that, both one single super defender and homogeneous defenders will defend perfectly, so measuring the minimum no longer makes sense. However, it is striking that the benefits of a heterogeneous team persist so close to that threshold, and the optimal ratio remains relatively stable over a wide range of speed 'budgets' in both settings.

Computational complexity of simulations: The results of the monotonicity-based computational acceleration discussed in Section 3.2 can be seen in Figure 5, corresponding to the simulations shown in Figure 3. As before, the left-hand column is the results for uniformly-random time attacks, and the right-hand column is the results for fixed time attacks, while the top row represents a single trial (corresponding to the top row of Figure 5) and the bottom row correspond to the average of $T = 200$ trials.

Each square is a 256×256 grid, representing the 256^2 combinations of speeds $\boldsymbol{v}$ for which we want to compute $\text{Opt}(\boldsymbol{v})$; the shade of a given point represents the fraction of times Alg. 1 had to be run on for that specific $\boldsymbol{v}$ (as opposed to being known already by monotonicity), running from yellow (Alg. 1 never run) to purple (Alg. 1 run). Note that because they represent a single trial (each), every point in the top two graphs takes a value of either 0 or 1.

We note a few things: (i) the savings increase strongly where $\mathbb{E}[\mathrm{Opt}(\boldsymbol{v})]$ is flatter (this is expected since $\nabla_{\boldsymbol{v}}\mathbb{E}[\mathrm{Opt}(\boldsymbol{v})]$ corresponds to the probability that there is a step at $\boldsymbol{v}$, and having a step nearby means the condition is less likely to be satisfied); (ii) there are darker points at regular intervals (such as in the center), which correspond to the combinations which are evaluated earlier.

Even with $m = 2$ and the strategic use of monotonicity, which can save up to about 95% of the running time, this can get big fairly quickly.

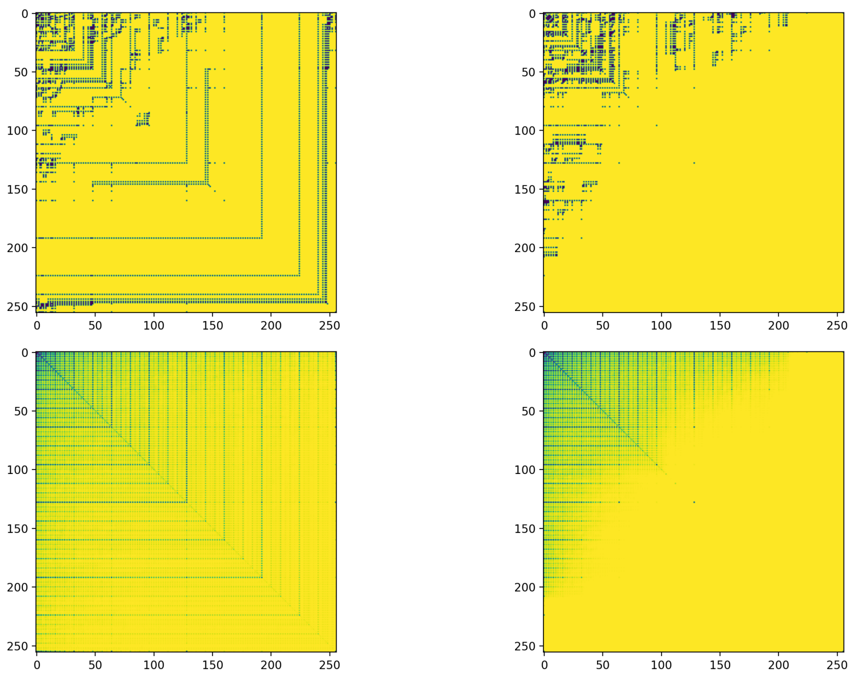

Fig. 5: Monotonicity savings for the trials depicted in Figure 3. Uniformly-random attack times on the left, and fixed attack times on the right. Axes labeled by position in the vector of possible speeds (0 to $g-1$). Top row is for one trial (corresponding to the single trials shown in Figure 3) and bottom is average over 200 trials.

5.2 Simulations for unit horizon

Simulation results for the case of two defenders on a circular perimeter with unit horizon are shown in Figure 6. Note that in this case, heterogeneity is not beneficial, it is even detrimental. The optimal speed allocation is to assign the entire speed budget to one defender or split it equally.

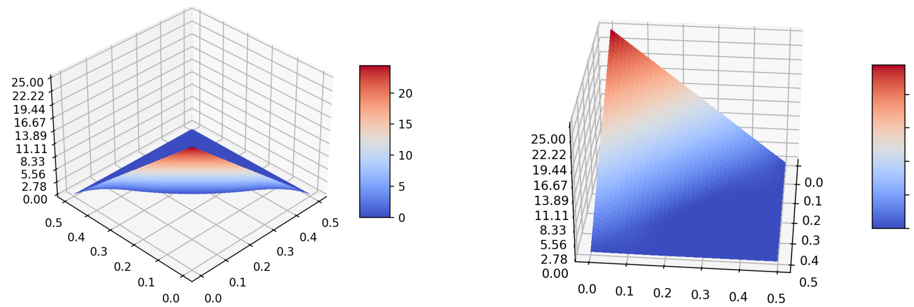

Fig. 6: Unit-horizon case, 2 defenders evaluated at $g = 128$ grains for speeds $(v_{\min}, v_{\max}] = (0, 0.5]$ for 10000 trials and $n = 25$ attacks. *Left:* back view, note the lack of the 'ridge' seen in Figure 3. *Right:* front view.

6 Conclusion

We introduced and studied a minimal model to map out how and why heterogeneity in robotic teams affects performance in perimeter defense applications.

On the one hand, we showed that a heterogeneous team achieves better performance when full information of the oncoming attacks is available to the defenders. Moreover, we uncovered a seemingly universal behavior, where the ratio of optimal defender speeds is nearly constant for a range of problem parameters.

On the other hand, we proved that heterogeneity is detrimental to the system's performance in the converse case where minimal attack information is available. These results suggest that heterogeneity is potentially a non-robust property, since less system information dramatically decreases its usefulness.

Future directions involve quantifying and studying the use of heterogeneity when intermediate levels of information are available to the defenders. This would explore the existence of a phase transition where heterogeneity changes from decreasing to improving system performance. Possible scenarios include varying the horizon length of incoming attacks between the cases of 1 and ∞ considered in the paper. Another scenario augments the unit time horizon with the knowledge of the number of remaining attacks. In particular, we conjecture that even in this case defenders should always capture attacks if possible and that heterogeneity remains detrimental. Lastly, we wish to perform numerical simulations for a larger number of defenders.

References

1. Adler, A., Mickelin, O., Ramachandran, R.K., Sukhatme, G.S., Karaman, S.: The Role of Heterogeneity in Autonomous Perimeter Defense Problems (2022), https://arxiv.org/abs/2202.10433
2. Casey, M.: Drones bring fight and flight to battle against poachers. Scientific American (online) **20** (2014)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, 2nd edn. (2001)

4. Fuchs, Z.E., Khargonekar, P.P., Evers, J.: Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game. In: Proc. IEEE Conf. Decis. Control (CDC). pp. 3091–3097 (2010)
5. Isaacs, R.: Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization. SIAM series in applied mathematics, Wiley (1965)
6. Lee, E.S., Shishika, D., Kumar, V.: Perimeter-defense game between aerial defender and ground intruder. In: Proc. IEEE Conf. Decis. Control (CDC). pp. 1530–1536 (2020)
7. Lykou, G., Moustakas, D., Gritzalis, D.: Defending airports from uas: A survey on cyber-attacks and counter-drone sensing technologies. Sensors **20**(12), 3537 (2020)
8. Mayya, S., Ramachandran, R.K., Zhou, L., Senthil, V., Thakur, D., Sukhatme, G.S., Kumar, V.: Adaptive and risk-aware target tracking for robot teams with heterogeneous sensors. IEEE Robot. Autom. Lett. (2022)
9. National Interagency Fire Center: Statistics. https://www.nifc.gov/fire-information/statistics, accessed: 2022-02-01
10. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons (2014)
11. Ramachandran, R.K., Preiss, J.A., Sukhatme, G.S.: Resilience by reconfiguration: Exploiting heterogeneity in robot teams. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6518–6525 (Nov 2019)
12. Ramachandran, R.K., Pierpaoli, P., Egerstedt, M., Sukhatme, G.S.: Resilient monitoring in heterogeneous multi-robot systems through network reconfiguration. IEEE Trans. Robot. **38**(1), 126–138 (2022)
13. Ravichandar, H., Shaw, K., Chernova, S.: Strata: unified framework for task assignments in large teams of heterogeneous agents. Autonomous Agents and Multi-Agent Systems **34** (2020)
14. Raytheon Missiles & Defense: Iron Dome System and SkyHunter Missile. https://www.raytheonmissilesanddefense.com/capabilities/products/irondome, accessed: 2022-02-01
15. Santos, M., Egerstedt, M.: Coverage control for multi-robot teams with heterogeneous sensing capabilities using limited communications. In: IEEE Int. Conf. Intell. Robots Syst. (IROS). pp. 5313–5319 (2018)
16. Shishika, D., Kumar, V.: Local-game decomposition for multiplayer perimeter-defense problem. In: Proc. IEEE Conf. Decis. Control. pp. 2093–2100 (2018)
17. Shishika, D., Kumar, V.: A review of multi agent perimeter defense games. In: Zhu, Q., Baras, J.S., Poovendran, R., Chen, J. (eds.) Decision and Game Theory for Security. pp. 472–485. Springer International Publishing, Cham (2020)
18. Shishika, D., Paulos, J., Dorothy, M.R., Ani Hsieh, M., Kumar, V.: Team composition for perimeter defense with patrollers and defenders. In: Proc. IEEE Conf. Decis. Control (CDC). pp. 7325–7332 (2019)
19. Shishika, D., Paulos, J., Kumar, V.: Cooperative team strategies for multi-player perimeter-defense games. IEEE Robot. Autom. Lett. **5**(2), 2738–2745 (2020)
20. Twu, P., Mostofi, Y., Egerstedt, M.: A measure of heterogeneity in multi-agent systems. In: 2014 Proc. Am. Control Conf. pp. 3972–3977 (2014)
21. Von Moll, A., Garcia, E., Casbeer, D., Suresh, M., Swar, S.C.: Multiple-pursuer, single-evader border defense differential game. J. Aerosp. Inf. Syst. **17**(8), 407–416 (2020)
22. Yan, R., Shi, Z., Zhong, Y.: Reach-avoid games with two defenders and one attacker: An analytical approach. IEEE Trans. Cybern. **49**(3), 1035–1046 (2019)