IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Path Defense in Dynamic Defender-Attacker Blotto Games (dDAB) with Limited Information

Austin K. Chen[1*], Bryce L. Ferguson[2*], Daigo Shishika[3], Michael Dorothy[4], Jason R. Marden[2], George J. Pappas[1], Vijay Kumar[1]

***Abstract*— We consider a path guarding problem in dynamic Defender-Attacker Blotto games (dDAB), where a team of robots must defend a path in a graph against adversarial agents. Multi-robot systems are particularly well suited to this application, as recent work has shown the effectiveness of these systems in related areas such as perimeter defense and surveillance. When designing a defender policy that guarantees the defense of a path, information about the adversary and the environment can be helpful and may reduce the number of resources required by the defender to achieve a sufficient level of security. In this work, we characterize the necessary and sufficient number of assets needed to guarantee the defense of a shortest path between two nodes in dDAB games when the defender can only detect assets within k-hops of a shortest path. By characterizing the relationship between sensing horizon and required resources, we show that increasing the sensing capability of the defender greatly reduces the number of defender assets needed to defend the path.**

I. Introduction

The emergence of new technologies in multi-robot systems and their applications in surveillance [1], pick-and-place [2], delivery [3], etc., have motivated a large area of research on determining how to delegate tasks to each robot [4], [5] and how to allocate robotic resources to different regions [6], [7]. With an increased understanding of how to perform this task assignment comes improvements in the overall operation of the system and the potential to complete jobs with less physical resources.

One particular area where multi-robot systems can offer new opportunities is in environments with adversarial operators [8]. The use of security or defense systems has been studied in many different contexts including cross-fire attacks in network-routing [9], [10], security against malicious groups [11], defending networks of sub-systems from multiple attackers [12], [13], and many more. Within each of these settings, the defender's ability to offer security guarantees in the face of unknown adversarial actions is hard [14] and depends on their knowledge of the system environment and the adversary's capabilities. In this work, we seek to understand how increasing information about the adversary and the environment can improve a defender's ability to provide security guarantees with limited resources.

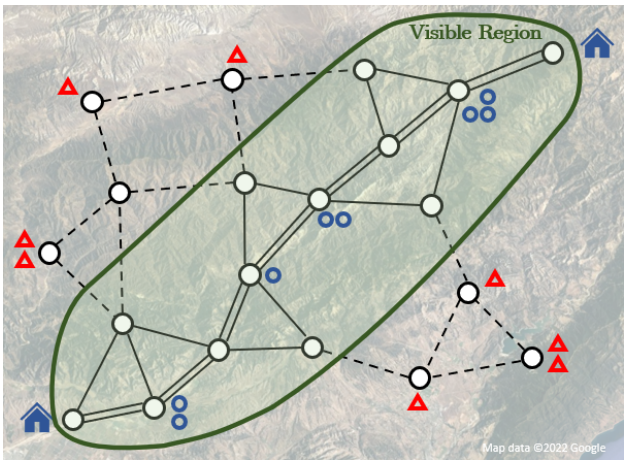


Fig. 1. Illustration of the limited-visibility path defense problem. At each node, the defender (blue) and the attacker (red) each posses a number of assets that they sequentially maneuver through the network. The objective of the defender is to guarantee each node on the double-line path has more defender assets than attacker assets at every time step. The defender can only detect attacker assets within its sensing horizon, represented as the shaded green region.

The interactions between a defender and an attacker (or adversary) can be described by a two-player zero-sum game, where each of the decision-makers' objectives is inversely aligned. One model that captures the key principles of these interactions are Colonel Blotto games [15], where two players each posses a finite reserve of troops that they allocate to various battlefields; at each battlefield, whichever player has allotted more troops wins the battlefield. The Colonel Blotto game has been used to develop algorithms and deploy security strategies in many domains including airport security, border control, and wildlife protection [16]–[18]. Additionally, researchers have used these games to study the interactions of defenders and attackers when there is incomplete information about the value of battlefields [19], [20] or the budget of the opposing colonel [21], [22]. Though these results provide a first glimpse at how information affects defender decision making, the results focus on each colonel's ability to win battlefields in a one-shot setting. Instead, we wish to investigate the conditions under which a defender may guarantee a defense objective against a dynamic adversary.

*The first two authors contributed equally as co-first authors.
[1]Austin, George, and Vijay are with the University of Pennsylvania {akchen, pappasg, kumar}@seas.upenn.edu
[2]Bryce and Jason are with the University of California, Santa Barbara {blferguson, jrmarden}@ece.ucsb.edu
[3] Daigo is with George Mason University dshishik@gmu.edu
[4] Michael is with DEVCOM Army Research Laboratory michael.r.dorothy.civ@army.mil

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Government, Department of Defense, or its components. We gratefully acknowledge the support from ARL Grant DCIST CRA W911NF-17-2-0181, NSF Grant CCR-2112665, ONR grants N00014-20-1-2822 and N00014-20-S-B001, and Lockheed Martin.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

In this work, we study the dynamic Defender-Attacker Blotto (dDAB) game where a defender and attacker sequentially maneuver assets in a network; on each node, whichever decision maker possesses more assets takes control of it. Originally introduced in [23], the dDAB game was used to address what strategies a team of robots can use to defend every node in a network from adversarial intervention. Here, we focus on the path guarding problem, where the defender must maintain control of each node in a shortest path between an origin and destination (Fig. 1). We then address how increased information about the adversary and the surrounding environment (in terms of visibility from the shortest path) affects the amount of defender resources required to obtain security guarantees. This work is closely connected to perimeter defense problems [24]–[27] and especially their multi-defender versions [28]–[30]; however, here we specifically focus on paths in networks and introduce a setting where the defender has limited information about the adversary.

For a defender with a limited sensing horizon (measured by how many hops from the chosen path the defender can detect an adversary, potentially using implemented surveillance equipment), we characterize the necessary and sufficient number of defender assets required to guarantee that the defender can maintain control of a shortest path between a start and a target node. We also provide an initial deployment and algorithm that realizes this guarantee. Our result shows that as the sensing horizon (and defender information) increases, the required number of assets to guarantee defense decreases.

II. Problem Formulation

In a finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we consider the problem of path defense, where a defender seeks to defend a predetermined path $\mathcal{P}$ from an adversary. We define the path $\mathcal{P}$ with cardinality $|\mathcal{P}|$ as a path graph with vertices $p_1, \dots, p_{|\mathcal{P}|}$ and $|\mathcal{P}|-1$ edges denoted by $(p_1, p_2), (p_2, p_3) \dots, (p_{|\mathcal{P}|-1}, p_{|\mathcal{P}|})$. We assume $|\mathcal{P}| \geq 3$ to avoid trivially short paths. Denote the start node S and the target T as the first and last nodes in the path respectively, so that $p_1 \equiv S$ and $p_{|\mathcal{P}|} \equiv T$. We will now make an assumption on the structure of $\mathcal{P}$, using the distance $d(v_1, v_2)$ to denote the minimum length of any path (as measured by the number of edges) between v_1 and v_2:

Assumption 1. *$\mathcal{P}$ is a shortest $S-T$ path in $\mathcal{G}$, i.e.*

$$d(S, T) = |\mathcal{P}| - 1. \tag{1}$$

Together, $\mathcal{P}$ and $\mathcal{G}$ define the environment in which a specific instance of the game is played. Since most of the operations and functions that follow depend on these parameters, we will omit the dependence on $\mathcal{P}$ and $\mathcal{G}$ when the relationship is clear.

The path defense game is played by a defender and an adversary. Both players have a finite amount of resources (or assets) at their disposal, denoted by $X \in \mathbb{R}_{>0}$ for the defender and $Y \in \mathbb{R}_{>0}$ for the adversary. Each player distributes their assets over the vertices of $\mathcal{G}$ in a deterministic and centralized fashion. The resulting asset distribution

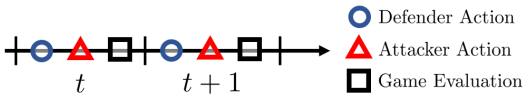


Fig. 2. Sequence of events at every step of the game. The defender first moves its assets based on the current adversary state, after which the adversary observes before making its own move. Finally, the game outcome is evaluated.

for each player is a point within the standard simplex of dimension $|\mathcal{V}|$, scaled so that the sum over elements is equal to the total resources of the player. For the defender, the asset distribution x is a vector defined as

$$x \in \left\{ z \in \mathbb{R}_{\geq 0}^{|\mathcal{V}|} \;\middle|\; \sum_{i=1}^{|\mathcal{V}|} z_i = X, z_i \geq 0 \right\}. \tag{2}$$

The adversary asset distribution y is defined as above with X replaced by Y. The scalar amounts of resources for the defender and adversary at vertex v are defined as x_v and y_v respectively.

The game terminates with an adversary win if the adversary has more assets than the defender on any of the path nodes, i.e. if $\exists v \in \mathcal{P}$ such that $y_v > x_v$. Otherwise, the path is defended. We define the *safe set*, $\mathcal{C}$, as follows:

$$\mathcal{C} = \{[x, y] \mid x_v \geq y_v, \forall v \in \mathcal{P}\}. \tag{3}$$

In this paper, we consider the dynamic version of the path defense game, where the game is played out over a series of timesteps t. Accordingly, the defender and adversary allocations become time-varying vectors $x(t)$ and $y(t)$ respectively. When the game begins, the defender and adversary select some initial states $x(0)$ and $y(0)$. We wish to consider only non-trivial starting conditions, i.e. we assume that $[x(0), y(0)] \in \mathcal{C}$. Because the adversary can choose any arbitrary $y(0)$, we will allow the defender to observe $y(0)$ before deciding its own initial state $x(0)$.

Given states $x(t)$ and $y(t)$, the game timestep t consists of the defender first transitioning its assets according to the function

$$x(t+1) = K^D(t) \cdot x(t), \tag{4}$$

where $K^D \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is a column stochastic state transition matrix (this enforces that the total number of defender assets remains unchanged over time). All elements of K^D must be nonnegative and entry $K^D_{i,j} > 0$ only if $(v_i, v_j) \in \mathcal{E}$. These constraints capture the notion that at every time step the defender can only move its assets up to one hop away from their current positions, and that assets may only be transferred along edges in $\mathcal{G}$.

In the same timestep t, after the defender transitions its assets, the adversary transitions its own assets according to

$$y(t+1) = K^A(t) \cdot y(t) \tag{5}$$

and the result of the game is evaluated. If $[x(t+1), y(t+1)] \notin \mathcal{C}$, then the game is terminated and declared an adversary win. Otherwise, the game continues to the next timestep $t+1$. This order of play is shown in Figure 2.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

We make no assumptions on how $K^A(t)$ is generated, but it must obey the graph and resource preservation constraints. Note that in this formulation, the problem takes the form of a Stackelberg game at every timestep since the adversary may observe the defender's action before taking its own. In contrast, the defender must generate its strategy without knowing where the attacker will move.

In this work, we are particularly interested in understanding the effect that information has on the defender's capability to maintain defense of the path. While both players know the state of their own assets and the adversary can observe the state of the defender's assets, the defender may not be able to fully observe adversarial assets. Let k be the defender's sensing distance and define the visible region $\mathcal{U}_k \subseteq \mathcal{V}$ as the set of vertices where the defender is able to observe the adversary's assets. If we define the minimum path distance function $d^*(v)$ for a vertex $v \in \mathcal{V}$ as

$$d^*(v) = \min_{p_i \in \mathcal{P}} d(v, p_i) \tag{6}$$

then the visible region $\mathcal{U}_k$ for a specific k is given as

$$\mathcal{U}_k = \{v \in \mathcal{V} \mid d^*(v) \leq k\}. \tag{7}$$

Since the defender can only observe adversary assets when they are in $\mathcal{U}_k$, the defender only observes a sub-vector of the adversary state $y(t)$. We call this sub-vector the observable adversary state $\widehat{y}(t)$ and define it as

$$\widehat{y}(t) = \{y_v(t) \mid v \in \mathcal{U}_k\} \tag{8}$$

where $y_v(t)$ is the adversary assets at vertex v during time t. We can now specify the form of $K^D(t)$ as

$$K^D(t) = \pi^D(x(t), \widehat{y}(t)) \tag{9}$$

where $\pi^D \colon \mathbb{R}^{2|\mathcal{V}|} \to \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the control policy of the defender.

In this paper, we are interested in investigating how to determine π^D and the minimum number of defender assets X required to ensure that the path is always defended, i.e. that $[x(t), y(t)] \in \mathcal{C} \ \forall \ t$. We will also investigate how the required amount of defender assets changes as a function of the sensing distance k.

A. Properties of Path Guarding Games

We make a few observations about the dDAB problem as specified by the preceding problem formulation. First, no vertex $v \in \mathcal{V} \setminus \mathcal{P}$ can be connected to a pair of path nodes $p_i, p_j \in \mathcal{P}$ such that $d(p_i, p_j) > 2$. This follows because of the assumption that $\mathcal{P}$ is a shortest path between S and T in $\mathcal{G}$, as if it were not true then a shorter path would exist through v. Additionally, from this property, we may also see that each node may be connected to at most 3 distinct path vertices, because otherwise the distance between the connected path vertices would exceed 2. In this case, the 3 path vertices must also be consecutive.

Another observation we make is that each vertex $v \in \mathcal{V}$ has its own static path distance $d^*(v)$. We may then organize the vertices in terms of their path distances. This also means that every visible region with a certain sensing distance is always a subset of a visible region with a greater sensing distance, i.e. $k \leq m \implies \mathcal{U}_k \subseteq \mathcal{U}_m$. Intuitively, this means that as the sensing distance increases, the corresponding visible region grows to include nodes that are farther from the path, e.g. for $k = 0$ the visible region is just $\mathcal{P}$, for $k = 1$ the visible region is $\mathcal{P}$ and all 1-hop neighbors of $\mathcal{P}$, etc.

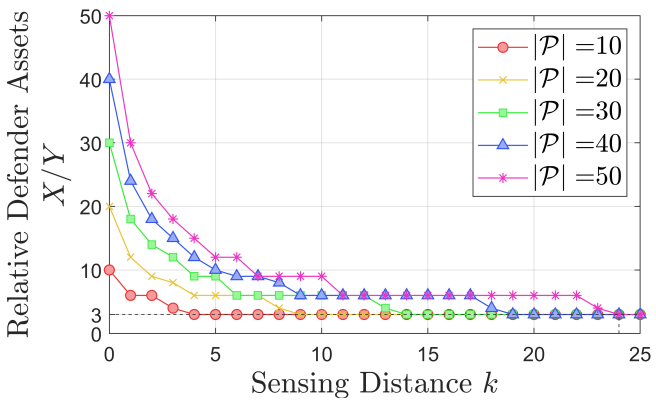


Fig. 3. Number of defending assets needed over various sensing distances. The number of defender assets needed to guarantee defense on a shortest path $\mathcal{P}$ decreases as the sensing horizon increases. When the sensing horizon is $k > |\mathcal{P}|/2$ only $3Y$ defender assets are needed.

III. Main Result

We seek to understand how additional information can aid the defender when operating in an uncertain environment against an adversary. In such settings, the environment (i.e. network structure) may influence how difficult it is for the defender to maintain their objective; however, it may be difficult or impossible for the defender to know this a priori. We therefore derive a scheme for defense policies that will succeed in any, considered network structure.

To understand how information can help a defender maintain their objective of securing $\mathcal{P}$, Theorem 1 characterizes the necessary and sufficient amount of defender assets needed to guarantee that $\mathcal{P}$ is defended against any adversarial strategies.

Theorem 1. *Let $\mathcal{P}$ be a path to be defended which satisfies the shortest path condition* (1) *in $\mathcal{G}$. For a given defender sensing horizon k, the condition*

$$X \geq \left(3\left\lfloor \frac{|\mathcal{P}|}{2k+3} \right\rfloor + \min\left\{\mathrm{mod}(|\mathcal{P}|, 2k+3), 3\right\}\right) Y \tag{10}$$

is sufficient for $\mathcal{P}$ to be defended in any graph $\mathcal{G}$ and necessary for $\mathcal{P}$ to be defended across all graph structures $\mathcal{G}$.

Here, $\mathrm{mod}(\diamond, \blacksquare)$ denotes the remainder of the Euclidean division of $\diamond$ by $\blacksquare$. The proof of Theorem 1 appears in Section IV with various parts separated into subsections.

In Fig. 3, we show how the number of assets needed, quantified by (10), decreases as the sensing horizon increases. Quantifying this improvement provides insights into the returns of investing in additional information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

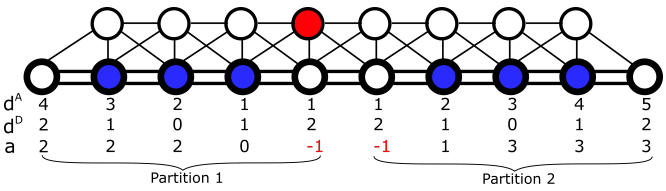


Fig. 4. Example illustrating the distance d^A from a single adversary, distance d^D from the partition's center defender, and calculated advantage a for each path node. Here, path nodes are shown by the bold circles at the bottom and are connected by double lines. Nodes with defender units are filled with blue, and the adversary's node is filled with red. Note that within each partition, defenders move as a platoon. In this example, $k = 1$ and so the partition size is $2 \cdot 1 + 3 = 5$. Since there are negative advantage values (highlighted in red), the defenders in each partition must move towards the center so as to prevent the adversary from winning the game.

We can further our understanding of the value of information by comparing (10) with the number of defender assets needed when there is no visibility constraint.

First, note that if the defender has full visibility (i.e., $k \to \infty$), then in any network $\mathcal{G}$ we have that $X \geq 3Y$ defender assets are necessary and sufficient to guarantee path guarding. This follows when the sensing distance is great enough, as for any sensing distance k such that $2k+3 > |\mathcal{P}|$, then $X = 3Y$ is necessary and sufficient from (10).

We can see that the defender need only have sensing distance $k = |\mathcal{P}|/2$ to guarantee defense with the same number of assets as the full visibility case. Accordingly, in Fig. 3 we see that number of needed defender assets quickly decreases and saturates for $k \geq |\mathcal{P}|/2$.

IV. Proof of Theorem 1

To prove Theorem 1, we start by assuming the adversary possesses a single, unsplittable asset; as such $Y = 1$ and X only takes integer values. We will show in Section IV-C that the results from this approach can be generalized to the case where any number of adversary assets can be split into multiple subgroups, including fractional assets.

The proof proceeds in two major parts. In subsection IV-A, we construct a strategy which shows that (10) is sufficient for guarding, and thus, an upper bound on the number of defenders required. In subsection IV-B, we show that (10) is necessary for guarding, and thus, a lower bound on the number of defenders required. Together, these results quantify the relationship between the sensing distance and the amount of resources required to defend $\mathcal{P}$. The proof of Theorem 1 relies on several lemmas, the proofs of these lemmas can be found in an online appendix.

A. Sufficient Algorithm

In order to show that (10) is sufficient, we present a defender algorithm which ensures that all nodes are defended. First, we split $\mathcal{P}$ into disjoint partitions of size $2k + 3$ each. There may be one partition smaller than $2k + 3$ if $\mathrm{mod}\,(|\mathcal{P}|, 2k+3) \neq 0$. For partition ω, we denote the center index as c_ω, i.e. p_{c_ω} is the center node of the partition. We call a *platoon* a group of three unit defenders in consecutive nodes within a partition, which always move together and maintain single spacing. The node index for the center of the platoon is termed l_ω, and therefore one unit of defenders will be distributed at each of $p_{l_\omega - 1}, p_{l_\omega}$, and $p_{l_\omega + 1}$. An example of this distribution is shown in Figure 4. For the special case where the partition is of size 1 or 2, there is one unit defender asset placed on each node within the partition, and the assets do not move. As a result, the nodes within this smaller partition are always guarded and so we do not consider this partition in our analysis below.

Algorithm 1: DefenderStep transition function.

Input: Adversary position n_A, current platoon location indices $L = \{l_1, \ldots, l_{|L|}\}$
Output: Updated platoon locations L

for *Each partition* $\{p_i, \ldots, p_j\}$ *with index* ω **do**
  Compute advantages $\{a_i, \ldots, a_j\}$
  Compute advantage frontier $\mathcal{F}$
  if $a_i > 0 \ \forall a_i \in \mathcal{F}$ **then**
    $l_\omega \longleftarrow l_\omega + \mathrm{sign}(c_\omega - l_\omega)$
  else if $\exists a_k \in \{a_i, \ldots, a_j\}$ *s.t.* $a_k < 0$ **then**
    $l_\omega \longleftarrow l_\omega + \mathrm{sign}(k - l_\omega)$
    $l_\omega \longleftarrow \max(\min(l_\omega, j-1), i+1)$
Return L

Given a path node p_i and the adversarial asset's node location v_A, define the minimum adversary distance d_i^A as

$$d_i^A = d(p_i, v_A). \tag{11}$$

Since the position of the defender's asset may vary over time, d_i^A may also change over time but we omit this dependence for notational brevity when considering a single timestep. If the adversary's asset is unobserverable then $d_i^A = \infty$. Similarly, define the minimum defender distance d_i^D as the minimum distance of the platoon center to a path node p_i within the platoon's partition, i.e.

$$d_i^D = d(p_i, l_\omega) \tag{12}$$

for p_i in partition ω. Then, we define the *advantage* a_i at path node p_i as

$$a_i = d_i^A - d_i^D. \tag{13}$$

Example values of d_i^A, d_i^D, and a_i can be found in Figure 4.

Lemma 1. *Assume that the defender uses platoons to defend each partition. It is necessary in any adversary winning configuration that* $a_i \leq -2$ *for some* $i \in [1, |\mathcal{P}|]$.

Proof. For the adversary to win, it must place its asset on some path node $p_i \in \mathcal{P}$, and therefore, $d_i^A = 0$.

We observe that $d_i^D \geq 2$, since otherwise p_i would be occupied by one of the three defender assets in the partition's platoon. Therefore $a_i \leq -2$. □

Corollary 1. *Since Lemma 1 describes a necessary condition for the adversary to win, a sufficient condition for the defender to successfully defend indefinitely is*

$$a_i \geq -1 \ \forall i \in [1, |\mathcal{P}|], \forall t. \tag{14}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: Initialization procedure.

Data: Adversary position n_A, platoon location indices $L = \{l_1, \ldots, l_{|L|}\}$

Observe n_A
Initialize L such that $l_\omega = c_\omega \ \forall \omega$
while $\exists a_i < -1$ **do**
  $L \longleftarrow$ DEFENDERSTEP(n_A, L)
Initialize platoons according to L

Defender Strategy: The defenders in each partition calculate the advantage values independently for the nodes in their own partitions to decide how to transition at each timestep. If the advantage value for any node in its partition is negative, then platoon moves towards that node along $\mathcal{P}$. Otherwise, the platoon considers the advantage values of all nodes (inclusive) between its own center and the closest partition boundary, which we call the set of frontier advantages $\mathcal{F}$. If every advantage value in $\mathcal{F}$ is positive, the platoon moves towards the middle of the partition. If neither of these conditions holds, then the defenders remain at their current position. Note that platoons are restricted to move within their partition, i.e. if partition ω consists of nodes p_i through p_j with $i < j$ then $l_\omega \in [i+1, j-1]$. This procedure, repeated once every timestep, is described in Algorithm 1.

Defender Initialization: To initialize the defender positions, the defender may simply repeat this procedure until all advantage values are greater than or equal to -1. Then, the resulting configuration is selected as the initial distribution. This approach is described in Algorithm 2.

With this strategy, the platoon only moves in one of two directions, either towards S or T. Because of this, it is not clear what should be done when negative advantage values appear on both sides of a platoon (i.e. if there is a negative advantage between the platoon and S, and also between the platoon and T). We will now show that this situation will never arise under the assumption that $\mathcal{P}$ is a shortest path.

Lemma 2. *Assume that $(\mathcal{P}, \mathcal{G})$ satisfies* (1)*. Then, for each partition, the defense strategy specified by Algorithm 1 will have negative advantage values on at most one side of the platoon within that partition. If a negative advantage value exists on one side of the platoon, the advantage values on the opposite side must all be positive.*

Proof. The proof can be found in the full-length version online at `https://arxiv.org/abs/2204.04176`. □

Corollary 2. *When a platoon moves one step towards a negative advantage value a_i, no negative advantage values will appear on the opposite side (from a_i) of the platoon within the partition.*

From Lemma 2, we know that when a platoon moves towards negative advantage values, the advantage values within the partition on the opposite side must all be positive. Since the defender can only change any advantage value by 1 with a single platoon move, the advantage values in question must be nonnegative after the move.

Lemma 3. *If the adversary moves out of $\mathcal{U}_k$ at timestep t, the platoons move back to the center of their partitions at timestep $t+1$.*

Proof. See full-length version online. □

Lemma 4. *Using Algorithm 2, the defender can achieve a starting configuration such that $a_i \geq -1 \ \forall \ i \in [1, |\mathcal{P}|]$ in finite time.*

Proof. See full-length version online. □

Since the proposed algorithm requires no more than the amount of resources specified by (10), we will now prove sufficiency for Theorem 1 by showing that the proposed algorithm guarantees path guarding.

Proof of sufficiency for Theorem 1. From Lemma 4 we know that initially $a_i \geq -1 \ \forall \ i \in [1, |\mathcal{P}|]$. During its turn, the defender can move towards the node with negative advantage value to restore the advantage values from -1 to 0 if the adversary is not at the end of the partition. This is possible since the negative advantage value only appears on one side of the platoon as shown in Lemma 2. Negative advantage values will also not appear on the opposite site of the platoon after the move, as stated in Corollary 2. Any adversary move can change the advantage value at each node by at most 1, and so the defender can repeat this procedure at every timestep with the same result.

For an adversary at the end of the partition, the platoon is not able to restore the advantage value for the edge of the partition to be nonnegative, as the platoon would have to move out of the partition to do so. However, in this case the partition is still guarded by one of the defender assets adjacent to the center of the platoon. The adversary is also not able to further decrease the advantage in this case, as either moving out of the partition or towards the platoon would increase the advantage value at the partition edge.

Note that if the adversary asset leaves $\mathcal{U}_k$, it does not have to reappear at the same node from which it left. However, it is always true that $d*(v_A) = k$ whenever the asset first reappears after leaving $\mathcal{U}_k$, and from Lemma 3 we know that the $d_i^D \leq k+1$, so $a_i \geq -1$. Therefore, the defender can guarantee that $a_i \geq -1$ for all time. As stated in Corollary 1, this means that the defender can guard $\mathcal{P}$ indefinitely. □

B. Necessary Defender Assets

In this subsection, we show that (10) is necessary to guarantee the defense of the path $\mathcal{P}$ when any additional graph structure can be realized outside of $\mathcal{P}$ such that no shorter path exists between S and T. In Section IV-A, we provided a sufficient algorithm where we chose to place defender assets only on $\mathcal{P}$; we now show that any defense strategy that allocates assets throughout $\mathcal{G}$ can be replicated by a defense strategy that allocates assets only on $\mathcal{P}$.

Lemma 5. *Consider a defender policy π^D. The defender can guarantee the same defense of the path $\mathcal{P}$ in the graph $\mathcal{G}$ with the same number of assets X, by using a policy $\hat{\pi}^D$*

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

that only has assets on $\mathcal{P}$ at each time step, i.e. $x_p(t) = 0$ if $p \notin \mathcal{P}$.

Next, we show a winning condition for the attacker based on the defender's starting configuration.

Lemma 6. *A winning attacker strategy exists in some graph $\mathcal{G}$ if there exist three consecutive nodes $p_{\alpha-1}, p_\alpha, p_{\alpha+1}$ in the path $\mathcal{P}$ which the defender cannot move an asset onto each of in k defender actions.*

Now, we prove that (10) is necessary to guarantee defense across all graph structures. Consider the case where the attacker starts outside of the region of visibility; because this is a valid strategy for the attacker, defending against it generates a necessary condition. Following Lemma 6: when $|\mathcal{P}| = 1$, one defending asset is necessary to defend $\mathcal{P}$, when $|\mathcal{P}| = 2$, two defending assets are necessary to defend $\mathcal{P}$, and when $|\mathcal{P}| = 3$, three defending assets are necessary to defend $\mathcal{P}$. The following remark will allow us to compare the necessary number of defender assets in different paths.

Remark 1. *If X defender assets are necessary to defend $\mathcal{P}$, then at least X defender assets are necessary to defend $\mathcal{P}'$ if $|\mathcal{P}'| > |\mathcal{P}|$.*

Thus, three defending assets are necessary for all $|\mathcal{P}| > 3$. Next, consider two necessary conditions for defending subsets of $\mathcal{P}$:

1) A node $p \in \mathcal{P}$ is defended only if it is within k-hops of a defender asset.
2) For $i \in \{1, \ldots, ||\}$, $\{p_i, p_{i+1}, p_{i+2}\}$ are guaranteed-defended only if nodes $\{p_{[i-k]_+}, \ldots, p_{i+k+2}\}$ are initially allocated 3 defender assets.

If the first were contradicted, an attacker asset could enter the region of visibility and reach node p_i in k attacker actions while the defender asset would require $k+1$ actions. If the second were contradicted, then the defender would not be able to move assets into the three-in-a-row configuration described in Lemma 6 over $\{p_i, p_{i+1}, p_{i+2}\}$ in k defender turns and the attacker could take one of these nodes.

Now, for some $\beta \in \{1, \ldots, L\}$ consider the task of deploying additional defender assets when the defender's deployment on $\{p_1, \ldots, p_\beta\}$ is already given. Regardless of how many, or where, the assets in $\{p_1, \ldots, p_\beta\}$ are initially deployed, they cannot reach node $p_{\beta+k+1}$ in k-hops, so (from necessary condition 1), $p_{\beta+k+1}$ is not defended by defender assets in $\{p_1, \ldots, p_\beta\}$, and (from necessary condition 2) there must be three assets deployed on nodes $\{p_{\beta+1}, \ldots, p_{\beta+2k+3}\}$, regardless of the deployment of assets on nodes $\{p_1, \ldots, p_\beta\}$. As such, because we can pick any $\beta \in \{0, \ldots, L-1\}$ any continuous sequence of nodes of length $2k+3$ must have at least three assets deployed on them. From this, a shortest path of length $|\mathcal{P}| = (2k+3)n$ must have $3n$ defender assets, as we can form n disjoint, connected sequences of nodes in $\mathcal{P}$ that must each possess 3 defending assets, or $X \geq \left(3\frac{L}{2k+3}\right) Y$.

Next, consider $|\mathcal{P}| = (2k+3)n+1$ and $\{p_1, \ldots, p_{(2k+3)n}\}$ are fully defended by $3n$ assets. From necessary condition 2, there must be 3 defending assets within the first $k+3$ nodes. From necessary condition 1, these assets cannot reach (and thus defend) any node further than p_{2k+3}, i.e., node p_{2k+3+1} cannot be reached (or defended) by the first 3 defender assets. If $n > 1$, from necessary condition 2, nodes $\{p_{k+3+1}, \ldots, p_{3k+6}\}$ must posses 3 defending assets, and from necessary condition 1, these assets cannot reach (and thus defend) any node further than $p_{2(2k+3)}$; so, node $p_{2(2k+3)+1}$ cannot be reached (or defended) by the first 6 defender assets. Following this logic for any $n \geq 1$, we can see that $3n$ assets that fully defend the first $(2k+3)n$ nodes cannot reach $p_{(2k+3)n+1}$ and one additional asset is needed.

Similarly, if $|\mathcal{P}| = (2k+3)n+2$ or $|\mathcal{P}| = (2k+3)n+3$, and the $\{p_1, \ldots, p_{(2k+3)n}\}$ are fully defended by $3n$ assets, then nodes $p_{(2k+3)n+1}$, $p_{(2k+3)n+2}$, and $p_{(2k+3)n+3}$ cannot be reached by the first $3n$ assets and 2 or 3 additional assets are needed respectively.

From Remark 1, when $|\mathcal{P}| \in \{(2k+3)n+3, \ldots, (2k+3)(n+1)\}$, the same number of defender assets $(3n+3)$ are necessary. Therefore, the necessary number of assets is as expressed in (10).

C. Generalization to Fractional Attacker Strategies

We now relax our assumption that attacker has an unsplittable unit mass of assets to extend the results of Section IV-A and Section IV-B to the general adversary problem. Since there are no restrictions on the number of assets that may pass through an edge, and since the motions of assets are not affected by the motions of other assets, we may simply consider a series of parallel games for each adversary asset. For example, if the adversary divides its assets in a 70:30 split between two nodes, the defender can respond by first splitting its assets (independently at each node) proportionally in a 70:30 split and then playing two independent games. The first 70% of defenders track the 70% of adversaries, and the second group of 30% defenders follow the 30% of adversaries. Additional splits by the adversary can prompt this defender response recursively. Since the graph is finite, the number of games the defender must play is also finite.

Because the attacker strategy of moving each of their assets in the same mass remains viable, the necessary number of defender assets remains the same and matches the sufficient number proposed by Section IV-A.

Finally, note that a split of adversarial assets in the unobservable region does not change this result. The defender naturally does not react to adversarial actions outside the unobservable region, and once the adversarial assets appear in the visible region path defense can be treated as before by allocating defender resources proportionally and playing multiple independent games.

V. Conclusions and Future Work

In this paper, we investigated the necessary and sufficient number of defenders required to defend a path on an arbitrary graph in the dDAB game as a function of the defender's sensing distance. We derived an expression which lower bounds the amount of defender resources required to defend

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.

a shortest path as a function of the sensing distance, and also described a defender strategy that guarantees guarding of the path while matching our lower bound. Together, these results quantify the relationship between the sensing distance and the resources required to defend a shortest path.

Future work will examine key variations to the dDAB game such as additional motion models for the players (e.g., allowing the adversary to move faster than the defender) as well as different objective-region types (e.g., cycles).

REFERENCES

[1] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos, "Multi-robot three-dimensional coverage of unknown areas," *The International Journal of Robotics Research*, vol. 31, no. 6, pp. 738–752, 2012.

[2] H. I. Bozma and M. E. Kalalioglu, "Multirobot coordination in pick-and-place tasks on a moving conveyor," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 4, pp. 530–538, 2012.

[3] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[4] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[5] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.

[6] B. L. Ferguson and J. R. Marden, "Robust Utility Design in Distributed Resource Allocation Problems with Defective Agents," *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1650–1655, dec 2021.

[7] J. R. Marden, "The Role of Information in Distributed Resource Allocation," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 654–664, 2017.

[8] N. Agmon, S. Kraus, and G. A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2339–2345.

[9] M. S. Kang, S. B. Lee, and V. D. Gligor, "The Crossfire Attack," in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 127–141.

[10] D. Grimsman, J. P. Hespanha, and J. R. Marden, "Stackelberg equilibria for two-player network routing games on parallel networks," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5364–5369.

[11] G. Brown, M. Carlyle, J. Salmerón, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, pp. 530–544, 2006.

[12] M. Abdallah, T. Cason, S. Bagchi, and S. Sundaram, "The Effect of Behavioral Probability Weighting in a Simultaneous Multi-Target Attacker-Defender Game," in *2021 European Control Conference (ECC)*, 2021, pp. 933–938.

[13] D. Kovenock and B. Roberson, "The optimal defense of networks of targets," *Economic Inquiry*, vol. 56, no. 4, pp. 2195–2211, 2018.

[14] F. Duvallet and A. Stentz, "Imitation learning for task allocation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3568–3573.

[15] B. Roberson, "The colonel blotto game," *Economic Theory*, vol. 29, no. 1, pp. 1–24, 2006.

[16] M. Tambe, *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

[17] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, 2008, pp. 125–132.

[18] L. Xu, S. Gholami, S. McCarthy, B. Dilkina, A. Plumptre, M. Tambe, R. Singh, M. Nsubuga, J. Mabonga, M. Driciru, and Others, "Stay ahead of Poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations (Short Version)," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 1898–1901.

[19] T. Adamo and A. Matros, "A Blotto game with incomplete information," *Economics Letters*, vol. 105, no. 1, pp. 100–102, 2009.

[20] D. Kovenock and B. Roberson, "A Blotto game with multi-dimensional incomplete information," *Economics Letters*, vol. 113, no. 3, pp. 273–275, 2011.

[21] K. Paarporn, R. Chandan, M. Alizadeh, and J. R. Marden, "A General Lotto game with asymmetric budget uncertainty," *arXiv preprint arXiv:2106.12133*, 2021.

[22] ——, "Characterizing the interplay between information and strength in Blotto games," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5977–5982.

[23] D. Shishika, Y. Guan, M. Dorothy, and V. Kumar, "Dynamic Defender-Attacker Blotto Game," *arXiv preprint arXiv:2112.09890*, 2021.

[24] D. Shishika and V. Kumar, "A review of multi agent perimeter defense games," in *International Conference on Decision and Game Theory for Security*. Springer, 2020, pp. 472–485.

[25] 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.

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

[27] E. S. Lee, D. Shishika, and V. Kumar, "Perimeter-defense game between aerial defender and ground intruder," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 1530–1536.

[28] 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)*, 2020, pp. 7971–7977.

[29] A. K. Chen, D. G. Macharet, D. Shishika, G. J. Pappas, and V. Kumar, "Optimal multi-robot perimeter defense using flow networks," in *Distributed Autonomous Robotic Systems*, F. Matsuno, S.-i. Azuma, and M. Yamamoto, Eds. Cham: Springer International Publishing, 2022, pp. 282–293.

[30] D. Shishika, D. Maity, and M. Dorothy, "Partial information target defense game," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8111–8117.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:34:53 UTC from IEEE Xplore. Restrictions apply.