Dynamic Adversarial Resource Allocation: A Complete Characterization

Yue Guan*
Georgia Institute of Technology
Atlanta, United States
yguan44@gatech.edu

Daigo Shishika*
George Mason University
Fairfax, United States
dshishik@gmu.edu

Jason R. Marden
University of California Santa Barbara
Santa Barbara, United States
jrmarden@ece.ucsb.edu

Panagiotis Tsiotras
Georgia Institute of Technology
Atlanta, United States
tsiotras@gatech.edu

Vijay Kumar
University of Pennsylvania
Philadelphia, United States
kumar@seas.upenn.edu

Abstract

This work studies a dynamic and adversarial resource allocation problem in a graph environment. A team of defender robots is tasked with protecting the environment from a team of attacker robots by ensuring the defender's numerical advantage at every node. The engagement is formulated as a discrete-time dynamic game which is referred to as the dynamic Defender Attacker Blotto (dDAB) game, where the two teams reallocate their robots in sequence and each robot can move at most one hop at each time step. The game terminates with the attacker's victory if any node has more attacker robots than defender robots. Our goal is to identify the necessary and sufficient number of defender robots to guarantee guarding. Through a reachability analysis, we derive the necessary and sufficient condition for a successful defense along with the associated strategies. Crucially, our result indicates that there is no incentive for the attacker team to split, which significantly reduces the search space for the attacker's winning strategies and also enables us to design defender counter-strategies using superposition. We develop an efficient numerical algorithm to identify the necessary and sufficient number of defender robots to defend a given graph. Finally, we present illustrative examples to verify the efficacy of the proposed framework.

Keywords

Multi-robot systems, Dynamic games, Resource allocation

ACM Reference Format:
Yue Guan*, Daigo Shishika*, Jason R. Marden, Panagiotis Tsiotras, and Vijay Kumar. 2023. Dynamic Adversarial Resource Allocation: A Complete Characterization. In *MSDM Workshop of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 8 pages.

1 Introduction

Deploying resources (robots, sensors, or supplies) to appropriate locations at the appropriate time is a fundamental problem in multi-agent systems, often studied as the multi-robot task allocation (MRTA) problem [5, 6]. In a real-world setting, resource allocation or MRTA must be performed in a dynamically changing environment. Among other factors, time-varying demand is one of the major sources of dynamics, exemplified by the research in wireless network [11], ride-sharing [2], and power-grid [1].

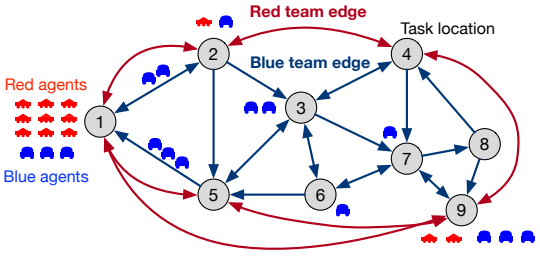


Figure 1: The adversarial resource allocation problem.

In this work, we study the dynamic resource allocation problem on a graph, where nodes represent physical locations and edges represent the traversability between those locations. The focus is on transporting the resources effectively in the environment to satisfy demands that change dynamically. Instead of achieving the desired allocation instantly, we require the robots to *traverse* through the environment. Such consideration arises naturally when dealing with embodied agents, such as robots or autonomous vehicles.

To stress the dynamic aspect of the problem, we consider demands that are generated by an adversary. More specifically, we formulate the problem as a dynamic (turn-based) game played between a blue team of defender robots and a red team of attacker robots. The defender team must ensure numerical advantage at every node the attacker robots are present. Whenever the attacker team has more robots in any node, the attacker team wins the game. In that sense, the demand imposed by the attacker team is a hard constraint that the defender team must continuously satisfy throughout the game. Many other safety-critical applications with dynamic demands (e.g., resilient power grid [1]) can be formulated as such hard-constrained resource allocation problems.

Our formulation also leads to feedback strategies that re-allocate resources based on the system state (demands generated by the attacker team and the current allocation of the defender team). The re-allocation will be done with all possible next actions of the opposing team in mind. This is a major difference with many prior works on resource allocation problems in the robotics community, where the focus has been either on achieving a desired terminal allocation

MSDM Workshop of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

that is fixed [3, 7], or on scheduling to satisfy a time-varying but known demand (e.g., multiple traveling salesman problem) [5, 10].

The main contributions of this work are: (i) formulation of a novel resource allocation problem that has high relevance to security applications; (ii) identification of the critical amount of resources that are necessary and sufficient to guarantee successful defense; (iii) derivation of the corresponding strategies that guarantee a successful defense; and (iv) the development of efficient algorithms to implement these strategies.

Due to space limitations, we only present proofs of certain theorems. The complete theoretical analysis will appear in a journal version of this work.

2 Problem Formulation

The dynamic Defender-Attacker Blotto (dDAB) game is played between two players: the defender and the attacker. The environment is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the N nodes represent locations, and the directed edges represent the players' traversability between those locations. To avoid degeneracy, we assume that $\mathcal{G}$ is strongly connected [3], i.e., every node is reachable from any other node. For notational simplicity, we assume that the two players share the same graph, but the present analysis easily extends to the case where the two players have different graphs.

A graph adjacency matrix $A \in \mathbb{R}^{N\times N}$ captures the traversability of the resources as follows:

$$[A]_{ij} = \begin{cases} 1 & \text{if an edge from } j \text{ to } i \text{ exists: } (j,i) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The out-degree of node i is denoted as $d_i = \sum_j [A]_{ji}$, and we denote the *out-neighbors* of node i as $\mathcal{N}_i = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$.

The total amount of resources for the defender and the attacker are denoted by $X \in \mathbb{R}_{>0}$ and $Y \in \mathbb{R}_{>0}$, respectively. The allocation of the defender's resources over the graph at time $t = 0, 1, \ldots$ is denoted by the state vector (allocation vector) $\mathbf{x}_t \in \mathbb{R}^N$, which lies on a scaled simplex, such that $[\mathbf{x}_t]_i \geq 0$ and $\sum_i [\mathbf{x}_t]_i = X$ for all $t = 0, 1, \ldots, T$ for some time horizon T. The state vector (allocation vector) $\mathbf{y}_t \in \mathbb{R}^N$ for the attacker also satisfies the same conditions with X replaced by Y. We use Δ_X and Δ_Y to denote the state space of the defender and the attacker. Note that we consider continuous resources ($\mathbf{x}_t$ and $\mathbf{y}_t$ are continuous variables).

The major difference from the original Colonel Blotto game is that the dDAB game is played over multiple time steps, and that the states evolve according to the following discrete-time dynamics:

$$\mathbf{x}_{t+1} = K_t \mathbf{x}_t \quad \text{and} \quad \mathbf{y}_{t+1} = F_t \mathbf{y}_t, \tag{2}$$

where K_t and F_t represent the *transition matrices* for the defender and the attacker, respectively. These matrices are left stochastic (column sum is unity), and the entries can take nonzero values only when the adjacency matrix has $[A]_{ij} = 1$. These matrices represent the action / control executed by the players. For example, an action K_t of the defender is admissible if and only if it satisfies the following *linear* constraints:

$$\sum_i [K_t]_{ij} = 1, \qquad \forall\, i \in \mathcal{V}, \tag{3}$$

$$[K_t]_{ij} \geq 0, \qquad \forall\, i, j \in \mathcal{V}, \tag{4}$$

$$[K_t]_{ij} = 0, \qquad \text{if } A_{ij} = 0. \tag{5}$$

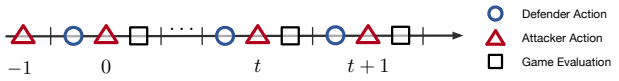


Figure 2: Sequence of events of the dDAB game.

We denote the admissible set for the matrices K_t as $\mathcal{K}$, which depends only on the underlying graph $\mathcal{G}$, and is time-invariant. The matrix F_t for the attacker also satisfies similar constraints, and we denote the set for all admissible matrices F_t as $\mathcal{F}$.

Similar to Colonel Blotto games [4, 8], the engagement at each location is modeled solely based on the amount of resources. Specifically, the defender successfully *guards* a location by allocating at least as many resources as the attacker does, whereas the attacker *breaches* a location by allocating more than what the defender does. For the dDAB game, the defender wants to prevent the attacker from breaching any location. Therefore, we define the terminal condition as

$$\exists\, i \in \mathcal{V}, \quad \text{such that} \quad [\mathbf{y}_t]_i > [\mathbf{x}_t]_i, \tag{6}$$

for some $t = 0, 1 \ldots, T$. The defender wins the game if it can prevent the attacker from winning if (6) for all $t = 0, 1, \ldots, T$. If (6) holds for all $T \geq 0$ then the defendercan prevent the attacker from winning indefinitely.

For the information structure, we assume that the players make decisions in sequence. Specifically, the defender acts first then the attacker acts next: i.e., the attacker can select its action after observing how the defender allocated its resources. The game outcome is evaluated after the attacker's move. To avoid the degenerate scenario where the attacker wins immediately in the first time step, we only specify $\mathbf{y}_{-1}$ as the initial condition. The game starts with the defender freely picking its distribution $\mathbf{x}_0$ after observing the attacker's initial state $\mathbf{y}_{-1}$. The timeline of the dDAB game is presented in Figure 2. In a realistic scenario, the two players make simultaneous actions. Therefore, our problem formulation provides a worst-case scenario for the defender.

An instance of dDAB game is defined by: (i) the available resources X and Y, and (ii) the underlying graph $\mathcal{G}$. Given a graph, our goal is to identify the necessary and sufficient number of resources for the defender to win the game. To formalize the goal above, we introduce the following multiplicative factor.

Definition 1 (Critical Resource Ratio)**.** *For a given (strongly-connected) graph $\mathcal{G}$ and a time horizon T, the CRR, $\alpha_T \geq 1$, is the* smallest *positive number such that, if*

$$X \geq \alpha_T Y, \tag{7}$$

then the defender has a strategy to defend till time step T against any admissible attacker strategy that starts at any initial state $\mathbf{y}_{-1} \in \Delta_Y$. We use α_∞ to denote the CRR that enables the defender to defend indefinitely.

The two main questions we address in this work are:

Problem 1. *Given a (strongly-connected) graph and a finite horizon T, what is the CRR α_T?*

Problem 2. *When $X \geq \alpha_T Y$, what is the corresponding defender strategy that guarantees defense over T time steps? When the defender does not have enough resources, what is the attacker strategy that ensures breaching?*

3 Reachable Sets and Required Sets

3.1 Reachable Sets

Since the dynamics of the two players are symmetric, we focus on the analysis of the defender's reachable sets and its action space $\mathcal{K}$. There are two major disadvantages working directly with the action space $\mathcal{K}$: (i) the higher dimensionality than the state space, i.e. $|\mathcal{E}| \gg |\mathcal{V}|$, and (ii) the nonuniqueness in the action that achieves a transition from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$. To avoid these issues, we directly consider the possible states that the defender can reach at the next time step.

Definition 2 (Reachable set from a single point)**.** *The reachable set from a single point $\mathbf{x}_t$, denoted as $\mathcal{R}(\mathbf{x}_t)$, is the set of all states that the defender can reach at the next time step with an admissible action. Formally, the reachable set from $\mathbf{x}_t$ is given by*

$$\mathcal{R}(\mathbf{x}_t) = \{\mathbf{x} \mid \exists K \in \mathcal{K} \text{ such that } \mathbf{x} = K\mathbf{x}_t\}. \tag{8}$$

Under the linear constraints in (3)–(5), the set of admissible actions, $\mathcal{K}$, is a bounded polytope in the $|\mathcal{E}|$-dimensional space. We use the extreme points (vertices) of this polytope to characterize $\mathcal{K}$.

Given the admissible action space $\mathcal{K}$, we define the set of *extreme actions* as

$$\hat{\mathcal{K}} = \left\{K \in \mathcal{K} \mid [K]_{ij} \in \{0, 1\}\right\}. \tag{9}$$

In words, $\hat{\mathcal{K}}$ contains all admissible actions K whose entries are either 0 or 1. The cardinality of $\hat{\mathcal{K}}$ is given by $\left|\hat{\mathcal{K}}\right| = \prod_{j\in\mathcal{V}} d_j$, where d_j is the out-degree of node j in the graph $\mathcal{G}$. We use ℓ to index the extreme actions in $\hat{\mathcal{K}}$, i.e. $\hat{\mathcal{K}} = \{\hat{K}^{(\ell)}\}_{\ell=1}^{|\hat{\mathcal{K}}|}$. The following theorem from our previous work [9] reveals the connection between the extreme actions and the admissible action set.

Theorem 1. *The extreme actions defined in* (9) *are the vertices of the polytope $\mathcal{K}$. Formally,*

$$\mathcal{K} = \mathrm{Conv}\big(\hat{\mathcal{K}}\big). \tag{10}$$

Consequently, for any admissible action $K \in \mathcal{K}$, there is a set of non-negative coefficients $\boldsymbol{\lambda} = \{\lambda^{(\ell)}\}_{\ell=1}^{|\hat{\mathcal{K}}|}$ such that $\sum_{\ell=1}^{|\hat{\mathcal{K}}|} \lambda^{(\ell)} = 1$ and such that

$$K = \sum_{\ell=1}^{|\hat{\mathcal{K}}|} \lambda^{(\ell)} \hat{K}^{(\ell)}. \tag{11}$$

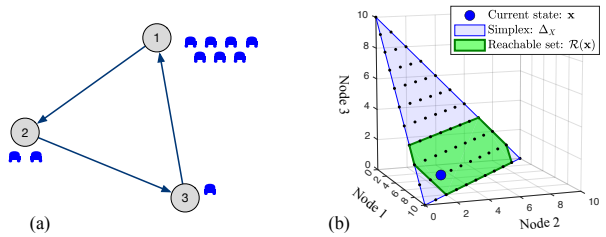
The extreme action set for the attacker is denoted as $\hat{\mathcal{F}}$; we use $\{\hat{F}^{(r)}\}_{r=1}^{|\hat{\mathcal{F}}|}$ to index its elements.

The reachable sets $\mathcal{R}(\mathbf{x}_t)$ are, in fact, also polytopes in Δ_X and it can be viewed as a transformation performed on the action space K. Formally, we have the following lemma [9].

Lemma 1. *Given a point $\mathbf{x}_t$, the reachable set $\mathcal{R}(\mathbf{x}_t)$ is a polytope given by $\mathcal{R}(\mathbf{x}_t) = \mathrm{Conv}\big(\{\hat{K}^{(\ell)}\mathbf{x}_t\}_{\ell=1}^{|\hat{\mathcal{K}}|}\big)$.*

Figure 3 presents an example of the reachable set for a three node graph. For discrete resources (robots) as illustrated in Figure 3(a), the defender is able to achieve any discrete state (black dots) that are contained in the reachable set.

Using the same argument for the attacker, we can obtain the polytope, $\mathcal{R}(\mathbf{y}_t)$ with vertices $\mathbf{w}_{t+1}^{(r)} = \hat{F}^{(r)}\mathbf{y}_t$ for $r = 1, 2, ..., |\hat{F}|$.



(a) (b)

Figure 3: Illustration of reachable set. (a) A directed graph with three nodes, each with a self-loop (omitted for clarity). (b) The defender's reachable set and its vertices.

Since any state in $\mathcal{R}(\mathbf{x}_t)$ can be reached at the next time step from $\mathbf{x}_t$, we will view this polytope as the action space for the defender at state $\mathbf{x}_t$. This definition of the action space resolves the two issues raised at the beginning of this section: dimensionality and nonuniqueness.

We further extend the definition of the reachable set of a single point to the reachable set of a polytope (potentially unbounded). The reachable set of a polytope will play a significant role in our later analysis of the optimal strategies for both players.

Definition 3 (Reachable set from a polytope)**.** *Given a polytope $P \subseteq \mathbb{R}^n_{\geq 0}$, the reachable set from this polytope, denoted as $\mathcal{R}(P)$, is the set of all states that the player can reach at the next time step with an admissible action starting from a state within P. Formally, the reachable set from P (for the defender) is given by*

$$\mathcal{R}(P) = \{\mathbf{x} = K\mathbf{x}_t \mid K \in \mathcal{K},\ \mathbf{x}_t \in P\}. \tag{12}$$

Lemma 2. *Given a polytope P, the reachable set $\mathcal{R}(P)$ is a polytope.*

3.2 Required Set

In this subsection, we identify the set of states $(\mathbf{x}_t, \mathbf{y}_t)$ that will immediately lead to termination in the next time step: i.e., for any defender action $\mathbf{x}_{t+1} \in \mathcal{R}(\mathbf{x}_t)$, the attacker has an action $\mathbf{y}_{t+1} \in \mathcal{R}(\mathbf{y}_t)$ to win the game by breaching at least one location at the next time step as in (6).

For the defender to defend every location, it is necessary and sufficient if the allocation vector, $\mathbf{x}$, matches or outnumbers $\mathbf{y}$ at every node i:

$$[\mathbf{x}]_i \geq [\mathbf{y}]_i \quad \forall i \in \mathcal{V}. \tag{13}$$

Since the attacker takes its action after observing the defender's action, the question is whether there exists $\mathbf{x}_{t+1} \in \mathcal{R}(\mathbf{x}_t)$ such that (13) is true for all $\mathbf{y}_{t+1} \in \mathcal{R}(\mathbf{y}_t)$. This observation leads to the following condition for guaranteed defense at $t+1$:

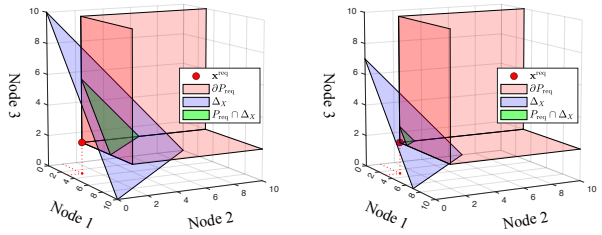
$$[\mathbf{x}_{t+1}]_i \geq \max_{\mathbf{y}_{t+1}\in\mathcal{R}(\mathbf{y}_t)} [\mathbf{y}_{t+1}]_i \quad \forall i \in \mathcal{V}. \tag{14}$$

Since $\mathcal{R}(\mathbf{y}_t)$ is a bounded polytope, for each node i the optimization $\max_{\mathbf{y}_{t+1}\in\mathcal{R}(\mathbf{y}_t)} [\mathbf{y}_{t+1}]_i$ can be viewed as a linear program, whose optimum is attained on one of the vertices of $\mathcal{R}(\mathbf{y}_t)$. Consequently, we define the minimum required resource at $t+1$ as $\mathbf{x}^{\text{req}}_{t+1}$, whose elements are given by

$$[\mathbf{x}^{\text{req}}_{t+1}]_i = \max_r \left[\mathbf{w}^{(r)}_{t+1}\right]_i, \tag{15}$$

where $\left\{\mathbf{w}^{(r)}_{t+1}\right\}_r = \left\{\hat{F}^{(r)}\mathbf{y}_t\right\}_r$ are the vertices of $\mathcal{R}(\mathbf{y}_t)$.

By allocating at least $[\mathbf{x}^{\text{req}}_{t+1}]_i$ to node i, the defender ensures that this node is defended against all feasible attacker actions at time



(a) (b)

Figure 4: Illustration of the required set $\mathcal{P}_{\text{req}}$ when $\mathbf{y} = [1, 0, 2]^\top$ and $\mathbf{x}^{\text{req}} = [3, 1, 2]^\top$. (a) Intersection between $\mathcal{P}_{\text{req}}$ and Δ_X when $X = 10$. (b) The case when $X = 6$.

step $t+1$. If the defender allocates $[\mathbf{x}_{t+1}]_i < [\mathbf{x}^{\text{req}}_{t+1}]_i$, then after observing defender's state (allocation), the attacker has a strategy $\mathbf{y}_{t+1} \in \mathcal{R}(\mathbf{y}_t)$ to win location i. Thus, $\mathbf{x}^{\text{req}}_{t+1}$ is the necessary and sufficient amount of resource for the defender to defend all locations at time $t+1$, given the current attacker resource distribution $\mathbf{y}_t$.

Notice that $X^{\text{req}}_{t+1} = \mathbf{1}^\top \mathbf{x}^{\text{req}}_{t+1}$ depends on $\mathcal{G}$ and $\mathbf{y}_t$. It is easy to see that the defender does not have a strategy to guarantee defense if $X^{\text{req}}_{t+1} > X$.

On the other hand, if $X^{\text{req}}_{t+1} \le X$, the defender can guarantee defense by selecting any $\mathbf{x}_{t+1}$ that is inside the polytope $\mathcal{P}_{\text{req}}(\mathbf{y}_t)$, which we call the *required set* defined as follows.

Definition 4. *Given the attacker's allocation $\mathbf{y}_t$ at time t, the required set for the defender at time $t+1$ is defined as:*

$$\mathcal{P}_{req}(\mathbf{y}_t) \triangleq \{\mathbf{x}_{t+1} \mid [\mathbf{x}_{t+1}]_i \ge [\mathbf{x}^{req}_{t+1}(\mathbf{y}_t)]_i,\ \forall\, i \in \mathcal{V}\}. \tag{16}$$

When $\mathcal{P}_{\text{req}}$ is given, Figure 4 illustrates how the intersection between $\mathcal{P}_{\text{req}}$ and the defender's state space changes as a function of defender's resource X.

4 No-Splitting Attacker

In this section, we will develop the tools to construct feedback strategies for the defender and the attacker. We first focus on the case where the attacker resources move as a single concentrated group (a blob). In Section 5, we will generalize the results to scenarios where the attacker splits its resource into multiple subgroups. Note that throughout this paper, we do not restrict the defender's allocation strategies.

Let $\mathbf{e}_i \in \mathbb{R}^n$ be the unit vector with its i-th element equal to one. In the sequel, we will use the shorthand $\mathbf{y}^{(i)} = Y\mathbf{e}_i$ to denote the attacker allocation that is fully concentrated on node i.

Definition 5 (No-splitting attacker strategy)**.** *A no-splitting attacker strategy concentrates all attacker resources at one node at each time t. That is, for all t, $\mathbf{y}_t = \mathbf{y}^{(i_t)} = Y\mathbf{e}_{i_t}$ for some $i_t \in \mathcal{V}$.*

4.1 K-step Safe Sets

To generate the defender strategy against *no-splitting* attacker strategies, we define the following k-step safe set, later referred to as the Q-set.

Definition 6 (k-step safe set)**.** *We define $\mathcal{Q}^{(i)}_k = \mathcal{Q}^{(i)}_k(\mathcal{G})$ to be the region in $\mathbb{R}^n_{\ge 0}$ such that for a no-splitting attacker state $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$, the defender's state $\mathbf{x}_t \in \mathcal{Q}^{(i)}_k$ is necessary and sufficient to defend against any no-splitting attacker strategy until time step $t+k$.*

Theorem 2. *The following recursive expression provides the k-step safe set:*

$$\mathcal{Q}^{(i)}_0 = \mathcal{P}_{req}(\mathbf{y}^{(i)}), \tag{17a}$$

$$\mathcal{Q}^{(i)}_k = \left\{\mathbf{x} \middle| \mathbf{x} \in \mathcal{P}_{req}(\mathbf{y}^{(i)}) \wedge \mathcal{R}(\mathbf{x}) \cap \mathcal{Q}^{(j)}_{k-1} \ne \varnothing\ \forall j \in \mathcal{N}_i\right\}, \forall\, k \ge 1. \tag{17b}$$

Proof. We break the proof into the following two lemmas, where Lemma 3 is for sufficiency and Lemma 4 is for necessity. □

Lemma 3 (Sufficiency of Q-sets)**.** *Let the Q-sets defined in* (17)*, and suppose the attacker starts with $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$. Then, by having $\mathbf{x}_t \in \mathcal{Q}^{(i)}_k$, the defender can defend at least until time step $t+k$.*

Proof. We do a proof by induction.

Base Case: When $k = 0$, we have $\mathbf{x}_t \in \mathcal{Q}^{(i)}_t = \mathcal{P}_{\text{req}}(\mathbf{y}^{(i)})$. From the construction of $\mathcal{P}_{\text{req}}(\mathbf{y}^{(i)})$, we know that for all $\mathbf{y}_t \in \mathcal{R}(\mathbf{y}_{t-1}) = \mathcal{R}(\mathbf{y}^{(i)})$, we have $\mathbf{x}_t \ge \mathbf{y}_t$ componentwise. Thus, the defender can guarantee defense at time t.

Inductive hypothesis: Suppose that having $\mathbf{x}_t \in \mathcal{Q}^{(i)}_k$ guarantees defense until time $t+k$ given $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$.

Induction: Given $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$, we let $\mathbf{x}_t \in \mathcal{Q}^{(i)}_{k+1}$. Under the no-splitting strategy, suppose that the attacker selects $\mathbf{y}_t = \mathbf{y}^{(j)}$, for some arbitrary $j \in \mathcal{N}_i$. The attacker cannot immediately win with this (or any other) action since the defender state $\mathbf{x}_t \in \mathcal{Q}^{(i)}_{k+1} \subseteq \mathcal{P}_{\text{req}}(\mathbf{y}_{t-1})$ guarantees defense at time step t. After observing $\mathbf{y}_t = \mathbf{y}^{(j)}$, we let the defender select its next state so that $\mathbf{x}_{t+1} \in \mathcal{R}(\mathbf{x}_t) \cap \mathcal{Q}^{(j)}_k$. This new selection is reachable since $\mathbf{x}_t \in \mathcal{Q}^{(i)}_{k+1}$ ensures that $\mathcal{R}(\mathbf{x}_t) \cap \mathcal{Q}^{(j)}_k \ne \varnothing$ (from (17b)). After the defender's action, we are at a situation where $\mathbf{y}_t = \mathbf{y}^{(j)}$ and $\mathbf{x}_{t+1} \in \mathcal{Q}^{(j)}_k$. From the inductive hypothesis, the defender can defend another k steps from this time on. The defender can thus defend until time step $t+k+1$. □

Lemma 4 (Necessity of Q-sets)**.** *Let the Q-sets defined in* (17)*, and suppose the attacker starts with $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$. If $\mathbf{x}_t \notin \mathcal{Q}^{(i)}_k$, the attacker can win the game before or at time step $t+k$.*

Proof. The proof of this lemma is constructed through a similar inductive argument and is thus omitted. □

Theorem 3. *All Q-sets are polytopes.*

The proof of Theorem 3 is delayed to Section 6, where we introduce additional tools to efficiently construct the Q-sets.

4.2 Indefinite Defense

As the recursive definition of the Q-sets in (17) can be viewed as an iterative algorithm, its fixed point(s) is therefore of great interest to study.

Definition 7 (Indefinite Safe Set)**.** *We define the indefinite safe sets $\mathcal{Q}^{(i)}_\infty \subseteq \Delta_X$ as follows*

$$\mathcal{Q}^{(i)}_\infty = \bigcap_{k \ge 0} \mathcal{Q}^{(i)}_k. \tag{18}$$

The following theorem formalizes the natural conjecture that indefinite safe sets guarantee an indefinite defense for the defender.

Theorem 4. *If $\{Q_\infty^{(i)}\}_{i\in\mathcal{V}}$ are nonempty, then $\mathbf{x}_t \in Q_\infty^{(i)}$ is the necessary and sufficient condition for the indefinite defense given that the attacker is at $\mathbf{y}_{t-1} = \mathbf{y}^{(i)}$.*

The conditions on the graph that guarantee convergence of the iterative algorithm in (17) as well as the conditions for the existence of such fixed point(s) is ongoing research.

4.3 Q-Set Propagation

The Q-set propagation process is described in Algorithm 1. The iterative Q-set construction process stops if the Q-sets converge. In this case, we can conclude that the defender can *indefinitely* defend against all no-splitting attacker strategies.

Algorithm 1: Q-Prop

Inputs: Graph $\mathcal{G}$, attacker total resource Y, max planning horizon $H_{\max}$;

Set $Q_0^{(i)} = \mathcal{P}_{\text{req}}(\mathbf{y}^{(i)}) = \mathcal{P}_{\text{req}}(Y\mathbf{e}_i)$ for $i \in \mathcal{V}$ and $k_\infty = \infty$;
for $k = 1$ *to* $H_{\max}$ **do**
  Construct $Q_k^{(i)}$ using (17b) for all $i \in \mathcal{V}$;
  if $Q_k^{(i)} = Q_{k-1}^{(i)}$ *for all* $i \in \mathcal{V}$ **then**
    $k_\infty = k - 1$;
    Return: $\{Q_k^{(i)}\}_{k\le k_\infty, i\in\mathcal{V}}$, k_∞;
  end
end
Return: $\{Q_k^{(i)}\}_{k\le K, i\in\mathcal{V}}$, k_∞

4.4 K-step Strategies

From the proof of Theorem 2, one already sees the strategies that the defender and the attacker would deploy under the assumption that the attacker starts on one node and does not split. The defender utilizes Algorithm 2 to select its initial allocation, which maximizes its guaranteed survival time given its total resource X and the observed attacker initial allocation $\mathbf{y}_{-1}$. For the rest of the game, the defender uses Algorithm 3 as its feedback strategy.

Algorithm 2: Initial Defender Allocation (No Splitting)

Inputs: Graph $\mathcal{G}$, defender total resource X, attacker total resource Y, attacker initial allocation $\mathbf{y}_{-1} = \mathbf{y}^{(i_{t-1})}$, planning horizon $H_{\max}$;

Construct Q-sets via Algorithm 1:
$\left(\{Q_k^{(i)}\}_{k\le\min\{H_{\max},k_\infty\},i\in\mathcal{V}}, k_\infty\right) \leftarrow$ Q-prop$(\mathcal{G}, Y, H_{\max})$;
$k_{\max,0} \leftarrow$
$\arg\max_k \left\{k \le \min\{H_{\max}, k_\infty\} | \Delta_X \cap Q_k^{(i_{-1})} \neq \varnothing\right\}$;
$\mathbf{x}_0 \leftarrow$ any element in $Q_{k_{\max,t}}^{(i_{-1})}$
Return: Q-sets $\{Q_k^{(i)}\}_{k\le\min\{H_{\max},k_\infty\},i\in\mathcal{V}}$, initial allocation $\mathbf{x}_0$, guaranteed survival time $k_{\max,0}$

The following two algorithms describe the attacker strategy under the no-splitting restriction. In particular, Algorithm 4 presents the initial allocation for the attacker, and Algorithm 5 provides the feedback attacker strategy at time steps $t \ge 1$. As we will show in the next section, the attacker has no incentive to split, i.e., if the attacker can win a dDAB game by splitting, it can also win the game without splitting. Consequently, the algorithms presented here are enough for the attacker.

Algorithm 3: Feedback Defender Strategy (No Splitting)

Inputs: Q-sets, current time step $t \ge 1$, defender allocation $\mathbf{x}_t$, observed attacker allocation $\mathbf{y}_t = \mathbf{y}^{(i_{t-1})}$, planning horizon $H_{\max}$;

$k_{\max,t} \leftarrow$
$\arg\max_k \left\{k \le \min\{H_{\max}, k_\infty\} | \mathcal{R}(\mathbf{x}_t) \cap Q_k^{(i_{t-1})} \neq \varnothing\right\}$;
$\mathbf{x}_t \leftarrow$ any element in $Q_{k_{\max,t}}^{(i_{t-1})}$;
Return: Next allocation $\mathbf{x}_t$, guaranteed survival time $k_{\max,t}$

Algorithm 4: Attacker Initial Allocation

Inputs: Graph $\mathcal{G}$, defender total resource X, attacker total resource Y, planning horizon $H_{\max}$;

Construct Q-sets:
$\left(\{Q_k^{(i)}\}_{k\le\min\{H_{\max},k_\infty\},i\in\mathcal{V}}, k_\infty\right) \leftarrow$ Q-prop$(\mathcal{G}, Y, H_{\max})$;
if $\exists k \le \min\{H_{\max}, k_\infty\}$ *and* $i \in \mathcal{V}$ *such that* $\Delta_X \cap Q_k^{(i)} = \varnothing$ **then**
  $k_{\min,-1} \leftarrow$
  $\arg\min_k \left\{k \le \min\{H_{\max}, k_\infty\} \mid \Delta_X \cap Q_k^{(i)} = \varnothing\right\}$;
  $i_{-1}^* \leftarrow$ any element in $\{i \mid \Delta_X \cap Q_{k_{\min,-1}}^{(i)} = \varnothing\}$;
else
  $k_{\min,-1} \leftarrow \infty$;
  $i_{-1}^* \leftarrow$ any element in node set $\mathcal{V}$;
end
Return: Q-sets $\{Q_k^{(i)}\}_{k\le\min\{H_{\max},k_\infty\},i\in\mathcal{V}}$, initial allocation $\mathbf{y}_{-1} = \mathbf{y}^{(i_{-1}^*)}$, guaranteed breach time $k_{\min,-1}$.

Algorithm 5: Feedback Attacker Strategy

Inputs: Q-sets, current time step t, observed defender allocation $\mathbf{x}_t$, planning horizon $H_{\max}$;

if $\exists k \le \min\{H_{\max}, k_\infty\}$ *and* $i \in \mathcal{V}$ *such that* $\mathbf{x}_t \notin Q_k^{(i)}$ **then**
  $k_{\min,t} \leftarrow$
  $\arg\min_k \left\{k \le \min\{H_{\max}, k_\infty\} \mid \mathbf{x}_t \notin Q_k^{(i)},\ i \in \mathcal{V}\right\}$;
  $i_t^* \leftarrow$ any element in $\{i \mid \mathbf{x}_t \notin Q_{k_{\min,t}}^{(i)}\}$;
else
  $k_{\min,t} \leftarrow \infty$;
  $i_t^* \leftarrow$ any element in $\mathcal{N}_{i_{t-1}}$;
end
Return: Next allocation $\mathbf{y}_t = \mathbf{y}^{(i_t^*)}$, guaranteed breach time $k_{\min,t}$.

5 Main Results

5.1 Generalization to Splitting Attacker

To extend the analysis on no-splitting strategies to the case with general strategies, we introduce the notion of subteams.

Definition 8 (Attacker Subteam). *We refer to the attacker resource allocated to each node as an* attacker subteam. *The size of the i-th attacker subteam (on node i) at time t is simply* $[\mathbf{y}_t]_i$.

In general, any attacker action can be viewed as a superposition of the subteam actions, which results in the splitting and merging of subteams into a new set of subteams.

We define the defender subteam as the subset of defender resource that can be deployed to defend against an attacker subteam.

Definition 9 (Defender Subteam). *The i-th defender subteam is defined based on the condition that it can guard against the i-th attacker subteam until some terminal time T:*[1]

$$\mathbf{x}_{t,T}^{(i)} \triangleq \frac{[\mathbf{y}_{t-1}]_i}{Y}\hat{\mathbf{x}}_t^{(i)}, \text{ where } \hat{\mathbf{x}}_t^{(i)} \in Q_{T-t}^{(i)}. \tag{19}$$

The first subscript, t, tracks the current time step, and the second subscript, T, tracks the expected terminal time.

Note that the Q-set in (17) is defined based on the full attacker team size Y, and $\hat{\mathbf{x}}$ is used to denote an element of the original Q-sets. On the other hand, a defender subteam is its scaled version according to the size of the corresponding attacker subteam.

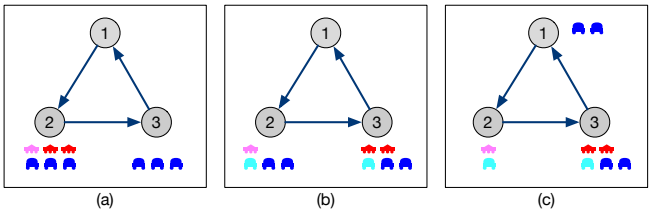


Figure 5: Splitting of an attacker subteam and the response of the corresponding defender subteam. Self-loops are omitted.

Figure 5(a) presents the initial state with a valid defender subteam placed against an attacker subteam at node 2. In Figure 5(b), the attacker resource splits into two: magenta with size 1 and red with size 2. Based on the observed attacker action, the original defender subteam also splits into two: cyan and blue which reacts against magenta and red, respectively. Figure 5(c) shows the reaction of each defender subteams (cyan and blue) against the corresponding attacker subteams.

We first present the main result of this section and provide the supporting lemmas later on.

Theorem 5. *Suppose that for a given terminal time T, and for some* $t \in \{0, 1, ..., T\}$, *the defender state can be described as a superposition of the subteams:*[2]

$$\mathbf{x}_t = \sum_i^n \mathbf{x}_{t,T}^{(i)}. \tag{20}$$

[1]For now, this condition only tells us that the defense will be guaranteed if the attacker subteam does not split further.

[2]This condition is implicitly dependent on $\mathbf{y}_{t-1}$ through the definition of the subteams in (19).

Then, the defender has a strategy to guarantee defense until time step T against any admissible attacker strategy.

Proof. We break the proof into three steps. **Step I:** In Lemma 5, we show that (20) is a sufficient condition for the defender to defend the current time step. **Step II:** Lemma 6 provides a strategy to maintain condition (20) at the next time step against any admissible attacker strategy. In other words, for $t \in \{0, \dots, T-1\}$, if $\mathbf{x}_t$ satisfies (20) for a given $\mathbf{y}_{t-1}$, then for any $\mathbf{y}_t \in \mathcal{R}(\mathbf{y}_{t-1})$, there is an admissible action K_t such that $\mathbf{x}_{t+1} = K_t\mathbf{x}_t$ satisfies the condition in (20) at $t+1$. **Step III:** Based on mathematical induction, condition in (20) is satisfied for all time steps. Therefore, the defense will be guaranteed until time step T. □

Lemma 5 (Safeness). *If the defender state satisfies* (20) *for some* $t \in \{0, 1, ..., T\}$, *then* $\mathbf{x}_t$ *is in the required set of* $\mathbf{y}_{t-1}$*:*

$$\mathbf{x}_t = \sum_i^n \mathbf{x}_{t,T}^{(i)} \quad \Rightarrow \quad \mathbf{x}_t \in \mathcal{P}_{req}(\mathbf{y}_{t-1}). \tag{21}$$

Proof. The proof is omitted due to page length limitations. □

Lemma 6 (Inductive condition). *Suppose the defender's state at time t satisfies*

$$\mathbf{x}_t = \sum_i^n \mathbf{x}_{t,T}^{(i)}. \tag{22}$$

Then, for any attacker action $\mathbf{y}_t \in \mathcal{R}(\mathbf{y}_{t-1})$, *there exists a defender's reaction* $\mathbf{x}_{t+1} \in \mathcal{R}(\mathbf{x}_t)$ *such that*

$$\mathbf{x}_{t+1} = \sum_i^n \mathbf{x}_{t+1,T}^{(i)}, \tag{23}$$

that is, the defender's state at the next time step can also be written as a combination of valid subteams defined in (19).

Proof. Denote the attacker's overall action that takes $\mathbf{y}_{t-1}$ to $\mathbf{y}_t$ as F_{t-1}. Then, this overall action F_t can be described as a combination of subteam actions as follows. Let $\boldsymbol{f}_i$ be the i-th column of F_{t-1}, i.e., $F_{t-1} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_n]$, where $\boldsymbol{f}_i^\top \mathbf{1} = 1$ (since F_{t-1} is left stochastic). We can interpret $\boldsymbol{f}_i$ to be the action of attacker subteam on node i at time $t-1$. More specifically, the fraction of a (possibly empty) subteam on node i relocating to node j is given by $[\boldsymbol{f}_i]_j$.

For notational convenience, we drop the second subscript, T, when denoting the defender subteams, $\mathbf{x}_{t+1,T}^{(i)}$. The defender subteam $\mathbf{x}_t^{(i)} = \frac{1}{Y}[\mathbf{y}_{t-1}]_i\hat{\mathbf{x}}_t^{(i)}$, where $\hat{\mathbf{x}}_t^{(i)} \in Q_{T-t}^{(i)}$, has the following strategy to react against the given splitting $\boldsymbol{f}_i$:

$$K^{(i)} = \sum_j [\boldsymbol{f}_i]_j K^{(i\to j)}, \tag{24}$$

where the action $K^{(i\to j)}$ corresponds to a satisficing defender action against a no-splitting attacker moving from node i to j, which provides $\hat{\mathbf{x}}_{t+1}^{(i\to j)} = K^{(i\to j)}\hat{\mathbf{x}}_t^{(i)} \in Q_{T-t-1}^{(j)}$.

Note that the subteam action $K^{(i)}$ can be interpreted as follows. First, the i-th defender subteam is divided into sub-subteams, according to the i-th attacker subteam's splitting action $\boldsymbol{f}_i$ from the previous time step (see Figure 5). The j-th defender "sub-subteam" of its i-th subteam then counteracts the j-th attacker sub-subteam that moves from node i to node j. This counteraction is achieved

by the defender sub-subteam applying the action $K^{(i\to j)}$. The next state achieved by the this defender sub-subteam is then given by

$$\mathbf{x}_{t+1}^{(i\to j)} = \frac{[f_i]_j [\mathbf{y}_{t-1}]_i}{Y} K^{(i\to j)} \hat{\mathbf{x}}_t^{(i\to j)} = \frac{[f_i]_j [\mathbf{y}_{t-1}]_i}{Y} \hat{\mathbf{x}}_{t+1}^{(i\to j)}.$$

Note that $\mathbf{x}_{t+1}^{(i\to j)}$ is only a part of the new j-th defender subteam, which originated from the previous i-th subteam. By collecting all defender resources from different subteams that reacted to the attacker resources that ended up at node j (i.e., the new j-th attacker subteam), the new j-th defender subteam can be computed as

$$\mathbf{x}_{t+1}^{(j)} = \sum_i \mathbf{x}_{t+1}^{(i\to j)} = \sum_i \frac{[f_i]_j [\mathbf{y}_{t-1}]_i}{Y} \hat{\mathbf{x}}_{t+1}^{(i\to j)}. \tag{25}$$

We now verify that this is a valid defender subteam, i.e., it can defend against $[\mathbf{y}_t]_j \mathbf{e}_j$ over the next $T-t-1$ time steps. By definition (19), the rescaled new j-th subteam is

$$\hat{\mathbf{x}}_{t+1}^{(j)} = \frac{Y}{[\mathbf{y}_t]_j} \mathbf{x}_{t+1}^{(j)} = \sum_i \frac{[f_i]_j [\mathbf{y}_{t-1}]_i}{[\mathbf{y}_t]_j} \hat{\mathbf{x}}_{t+1}^{(i\to j)}. \tag{26}$$

Noting that $\sum_i [f_i]_j [\mathbf{y}_{t-1}]_i = [\mathbf{y}_t]_j$, we see that $\hat{\mathbf{x}}_{t+1}^{(j)}$ is a convex combination of the states $\{\hat{\mathbf{x}}_{t+1}^{(i\to j)}\}_i$. Since Q-sets are polytopes (see Theorem 3) and $\hat{\mathbf{x}}_{t+1}^{(i\to j)} \in Q_{k-1}^{(j)}$ for all $i \in \mathcal{V}$, we can conclude that $\hat{\mathbf{x}}_{t+1}^{(j)} \in Q_{k-1}^{(j)}$. □

Corollary 1 (No Incentive to Split). *For a given graph $\mathcal{G}$ and resources X and Y, the attacker has a strategy to win the dDAB game if and only if it has a no-splitting winning strategy.*

Following the proof of Theorem 5, one can easily generalize the defender strategy in Section 4 to the scenarios where the attacker splits its resource over multiple nodes. The construction of the strategy is omitted due to space limit.

6 Algorithmic Solution

In this section, we first develop an algorithm to numerically construct the Q-sets. Recall the recursive definition of the Q-sets in (17b):

$$Q_k^{(i)} = \left\{ \mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{P}_{\text{req}}(\mathbf{y}^{(i)}) \wedge \mathcal{R}(\mathbf{x}) \cap Q_{k-1}^{(j)} \neq \varnothing \; \forall j \in \mathcal{N}_i \right\}.$$

We refer to the set $\{\mathbf{x} | \mathcal{R}(\mathbf{x}) \cap Q_k^{(j)} \neq \varnothing\}$ as the inverse reachable set, which consists of states from which the defender can reach $Q_k^{(j)}$ in the next time step. Formally,

Definition 10 (Inverse Reachable Set). *Given a set $P \subseteq \mathbb{R}^N_{\geq 0}$, we define the inverse reachable set of P as*

$$\mathcal{R}^{-1}(P) = \left\{ \mathbf{x} \;\middle|\; \mathcal{R}(\mathbf{x}) \cap P \neq \varnothing \right\}. \tag{27}$$

With the notion of the inverse reachable set, we can simplify the recursive construction of Q-sets in (17) as

$$Q_k^{(i)} = \left(\bigcap_{j \in \mathcal{N}_i} \mathcal{R}^{-1}(Q_{k-1}^{(j)}) \right) \cap \mathcal{P}_{\text{req}}(\mathbf{y}^{(i)}), \forall\, k \geq 1. \tag{28a}$$

To construct the reverse reachable sets, we introduce the notion of a reversed graph, which has the same node set as the original graph but with all the edges reversed.

Definition 11. *Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with connectivity matrix A. The reversed graph $\widetilde{\mathcal{G}} = (\mathcal{V}, \widetilde{\mathcal{E}})$ has the connectivity matrix $\widetilde{A}$ such that $\widetilde{A} = A^\top$.*

We denote $\widetilde{\mathcal{K}}$ as the admissible action set of $\widetilde{\mathcal{G}}$. The reachable set for the reversed graph is then defined as

$$\widetilde{\mathcal{R}}(\mathbf{x}) = \left\{ \mathbf{x}' \mid \exists \widetilde{K} \in \widetilde{\mathcal{K}} \text{ such that } \mathbf{x}' = \widetilde{K}\mathbf{x} \right\}.$$

Lemma 7. *For any graph $\mathcal{G}$, we have*

$$\mathcal{R}^{-1}(P) = \widetilde{\mathcal{R}}(P) \qquad \text{for all } P \subseteq \mathbb{R}^N_{\geq 0}.$$

Recall that reachable set of a polytope is also a polytope. With the equivalent Q-set definition in (28), one can easily show through an inductive argument that all Q-sets are polytopes, which is the statement of Theorem 3. Furthermore, the above result enables to efficiently propagate the Q-sets using the reachable sets of the reversed graph.

7 Numerical Illustrations

This section provides numerical examples that illustrate the theorems and algorithms developed in the previous sections.

7.1 Q-set Propagation

Figure 6 illustrates how the Q-sets, $Q_k^{(i)}$, changes with the horizon k. For the three-node graph selected for this example, the propagation in Algorithm 1 converges after four time steps, at which point the algorithm finds that $k_\infty = 4$. The CRR for this graph is $\alpha_\infty = 3$.

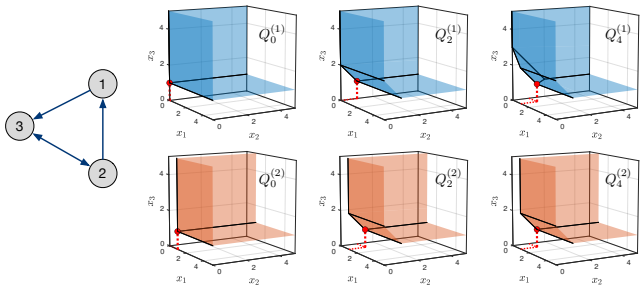


Figure 6: Examples of the Q-set propagation.

7.2 Effect of Edges on CRR

The relationship between the CRR and the graph structure is not straightforward. One might, for example, expect a positive correlation between the number of edges and the CRR, since an increase in the number of outgoing edges from a node increases the number of neighboring nodes that must be covered by the defender. However, we show by a counter-example (found by the algorithm) that this is not the case.

Figure 7 provides directed graphs with five nodes. The corresponding CRR α_∞ for each graph is obtained using Algorithm 1. In the simplest case of the ring graph, the defender only needs the same amount of resources as the attacker to guarantee indefinite defense, which matches the results in [9]. Interestingly, by adding only one directed edge connecting node 3 and node 4 to the graph,

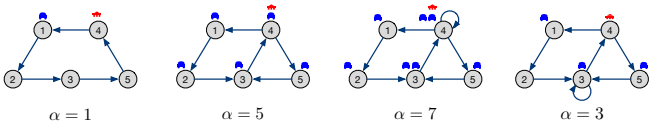


Figure 7: Examples of CRR under different graph structures.

the CRR changes drastically to $\alpha_\infty = 5$. By further adding a self-loop on node 4, the CRR goes up to seven. Notably, this is greater than the number of nodes on this graph. However, if we add a self-loop to node 3 instead of node 4, the CRR α_∞ decreases from 5 to 3, which implies that additional edges may benefit the defender as well.

7.3 Non-integer Resource Ratio

A natural conjecture regarding the CRR is that it is always integer-valued. Through the following dDAB game over a six-node graph in Figure 8, we show that it is possible that the resource ratio is a fraction for finite-horizon dDAB games. One can verify that three units of defender resources is not enough for a two-step defense. However, we will demonstrate that three and a half units is enough.

A game tree is presented in Figure 8 and different blue colors are introduced only to better visualize the splitting and regrouping of the defender resources. It is easy to verify that the initial defender state is in the $\mathcal{P}_{\text{req}}$. Now, the attacker has three feasible moves to make at time step 0: move to node 2, move to node 5 or stay at node 3. We only present the first two moves in Figure 8, since for the third move, the defender can just maintain its current state as a counter-measure and does not lose any survival time. Furthermore, we will focus on explaining the move to node 2 due to space limitations. After observing that the attacker moves to node 2, the defender takes action (a)[3] . The attacker then has two options, either move to node 5 or to node 6. Suppose the attacker moves to node 6, the defender initiates action (b)[4], which ensures that the configuration at the beginning of time step 2 is still in the required set. Similar moves can be made for trajectory (ii) to ensure the defense till the end of time step 2.

Through splitting and re-grouping its resources, the defender achieves defense with only an additional half unit of resource. The strategy presented is found by the algorithms introduced in Section 4, which verifies the efficacy of the proposed approach.

8 Conclusion

In this work, we formulated a dynamic adversarial resource-allocation problem by extending the Colonel Blotto game with ideas from population dynamics on graphs. Instead of achieving a desired allocation instantly, we require the resources of each player to traverse through the edges of the graph. An efficient reachable-set approach is developed to predict the evolution of the player's states. We provide a full characterization of the game by deriving the necessary and sufficient condition (the Q-sets) for either of the player to win

[3]Action (a) splits defender resources so that the one unit of defender on node 2 moves to node 6; half unit on node 3 moves to node 2 and the other half stays on node 3; the half unit on node 6 moves to node 1, and finally the one unit on node 5 stays.
[4]Action (b) moves the half unit on node 1 to node 5; the half unit on node 2 to node 5, the one unit on node 6 to node 1, and the rest of the resources on nodes 3 and 5 stay.

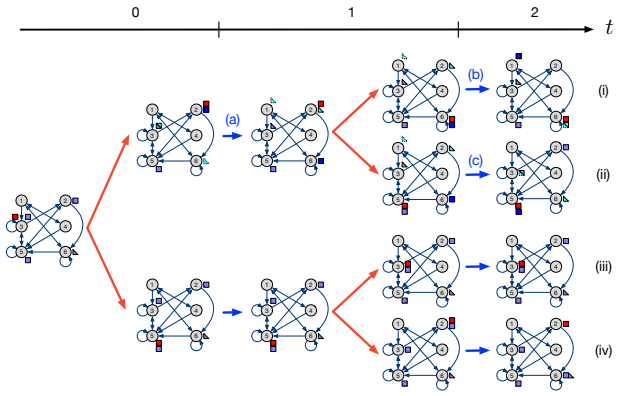


Figure 8: A two-time step game tree starting with one unit of attacker and three and a half unit of defender.

the game and the associated strategies to achieve that. The efficacy of the proposed approach is verified through numerical simulations.

References

[1] Aditya Ashok, Manimaran Govindarasu, and Jianhui Wang. 2017. Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proc. IEEE* 105, 7 (2017), 1389–1407.

[2] Xiaohui Bei and Shengyu Zhang. 2018. Algorithms for trip-vehicle assignment in ride-sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[3] Spring Berman, Adám Halász, M Ani Hsieh, and Vijay Kumar. 2009. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics* 25, 4 (2009), 927–937.

[4] Oliver Gross and Robert Wagner. 1950. *A Continuous Colonel Blotto Game.* Technical Report. RAND Corporation.

[5] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. 2015. Multi-robot task allocation: A review of the state-of-the-art. *Cooperative Robots and Sensor Networks 2015* (2015), 31–51.

[6] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. 2013. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32, 12 (2013), 1495–1512.

[7] Amanda Prorok, M Ani Hsieh, and Vijay Kumar. 2017. The impact of diversity on optimal control policies for heterogeneous robot swarms. *IEEE Transactions on Robotics* 33, 2 (2017), 346–358.

[8] Brian Roberson. 2006. The colonel Blotto game. *Economic Theory* 29, 1 (2006), 1–24.

[9] Daigo Shishika, Yue Guan, Michael Dorothy, and Vijay Kumar. 2022. Dynamic defender-attacker blotto game. In *2022 American Control Conference (ACC)*. IEEE, 4422–4428.

[10] Veniamin Tereshchuk, John Stewart, Nikolay Bykov, Samuel Pedigo, Santosh Devasia, and Ashis G Banerjee. 2019. An efficient scheduling algorithm for multi-robot task allocation in assembling aircraft structures. *IEEE Robotics and Automation Letters* 4, 4 (2019), 3844–3851.

[11] Yongjun Xu, Guan Gui, Haris Gacanin, and Fumiyuki Adachi. 2021. A Survey on Resource Allocation for 5G Heterogeneous Networks: Current Research, Future Trends, and Challenges. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 668–695. https://doi.org/10.1109/COMST.2021.3059896