

Game-Theoretic Drone Swarm Defense

A Case Study in Applied Differential Game Theory

Ross E. Allen

MIT Lincoln Laboratory

August 2026

Executive Summary

This technical report is a study of the use of differential game (DG) theory to solve the target-assignment and midcourse guidance problems of drone swarms tasked with intercepting opposing swarms in defense of high-value assets. The game-theoretic tactics—which treat the intruder swarm as a rational agent and seek a Nash equilibrium between defenders and intruders—are compared against baseline tactics that model the defense problem as a unilateral optimization of the defenders’ maneuvers. Monte Carlo simulation and Bayesian analysis show that the game-theoretic approach has a higher probability of successfully intercepting all intruders than the baseline techniques. This improvement in successful defense probability is most pronounced when the intruder swarm is capable of evasive maneuvers: relative to baseline optimization tactics, differential-game tactics increase estimated defense success from 94.6% to 96.8%, closing approximately 41% of the remaining gap to perfect defense. To add statistical credibility to this result, a paired-trial Bayesian analysis assigns a 99.9% posterior probability that differential-game tactics have a higher probability of successful asset defense than baseline tactics in this scenario.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Under Secretary of War for Research and Engineering under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of War for Research and Engineering.

© 2026 Massachusetts Institute of Technology.

Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Contents

1	Problem Formulation	3
1.1	Teams, State, and Information	3
1.2	Objectives and Solution Concepts	4
2	Guard/Defender Tactics	4
2.1	Baseline Tactics	5
2.1.1	Heading Error Midcourse Guidance	6
2.2	Differential Game Tactics	6
2.2.1	Dynamic Target Soft-Assignment	6
2.2.2	Game Theoretic Midcourse Guidance	8
2.2.3	Terminal Guidance	9
3	Simulation Environment	10
3.1	Interception Adjudication	10
3.2	Bandit/Intruder Tactics	10
4	Experiment Design	11
4.1	Monte Carlo Trials	11
4.2	Performance Metrics & Bayesian Analysis	11
5	Results & Discussion	13
6	Limitations & Future Work	18
A	Appendices	19
A.1	Discounted-Redundancy Assignment Reformulation	19

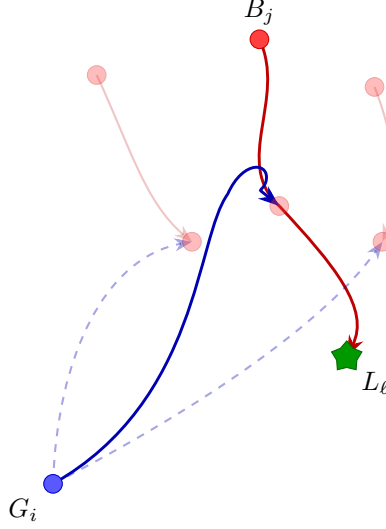


Figure 1: Conceptual drone swarm defense problem posed as a many-agent Lady-Bandit-Guard problem [7]. The guard drone G_i , which is one of a large set of guard drones, attempts to intercept a particular bandit drone B_j before it reaches a high-value asset L_ℓ , while considering the trade-offs of potentially pursuing different bandits.

1 Problem Formulation

The drone swarm defense problem is posed as a many-agent variant of the Lady-Bandit-Guard problem described by Rusnak [7]. A team of N_G guard drones (also referred to as *defenders* or *interceptors*) is tasked with intercepting an incoming swarm of N_B bandit drones (also referred to as *intruders* or *threats*) before any of them reach the set of N_L high-value assets (HVAs, called “ladies” in Rusnak’s terminology [7]).

The problem is multi-faceted, generally requiring optimization of both the assignment of specific guards to intercept specific bandits and guidance laws that generate dynamically-feasible trajectories that achieve those interceptions. Fig. 1 gives a conceptual illustration of an engagement with a small number of drones.

1.1 Teams, State, and Information

Let $G \triangleq \{0, \dots, N_G - 1\}$, $B \triangleq \{0, \dots, N_B - 1\}$, and $L \triangleq \{0, \dots, N_L - 1\}$ index the guards, bandits, and high-value assets (HVAs), respectively. The total population sizes N_G , N_B , and N_L are fixed for an episode. At any time, $G_A \subseteq G$ and $B_A \subseteq B$ denote the active guard and bandit index sets, with $N_{G_A} \triangleq |G_A|$ and $N_{B_A} \triangleq |B_A|$. Inactive vehicles remain represented in the full environment state but are omitted from active-roster planning problems.

Guards are modeled as fixed-speed, planar Dubins vehicles. For guard $i \in G$, the state, control, and equations of motion are

$$\begin{aligned} \mathbf{p}_{G,i} &= [p_{x,G,i} \quad p_{y,G,i}]^\top, \quad \mathbf{x}_{G,i} = [\mathbf{p}_{G,i}^\top \quad \psi_{G,i}]^\top, \\ u_{G,i} &= \omega_{G,i} \in \mathcal{U}_G, \quad \mathcal{U}_G \triangleq [-\bar{\omega}_G, \bar{\omega}_G], \\ \dot{\mathbf{x}}_{G,i} &= \begin{bmatrix} V_G \cos \psi_{G,i} \\ V_G \sin \psi_{G,i} \\ \omega_{G,i} \end{bmatrix}. \end{aligned} \tag{1}$$

Here $\mathbf{p}_{G,i}$ is planar position, $\psi_{G,i}$ is heading, V_G is the guard-team constant forward speed, and $\bar{\omega}_G$ is the maximum turn rate. The corresponding inertial velocity is $\mathbf{v}_{G,i} = V_G[\cos \psi_{G,i}, \sin \psi_{G,i}]^\top$. Bandits are defined analogously by replacing (G, i) with (B, j) and using V_B , $\bar{\omega}_B$, and $u_{B,j}$. Thus each vehicle state is three-dimensional and each control is a scalar heading rate. Stacking the dynamics over both teams defines $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}_G, \mathbf{u}_B)$. We ignore altitude changes and assume $V_G > V_B$; the rationale for excluding faster-bandit cases is discussed in Sec. 6.

The full-team and active-roster guard stacks are

$$\begin{aligned}
\mathbf{x}_G(t) &\triangleq [\mathbf{x}_{G,0}(t)^\top \cdots \mathbf{x}_{G,i}(t)^\top \cdots \mathbf{x}_{G,N_G-1}(t)^\top]^\top, \\
\mathbf{u}_G(t) &\triangleq [u_{G,0}(t) \cdots u_{G,i}(t) \cdots u_{G,N_G-1}(t)]^\top, \\
\mathbf{x}_{G_A}(t) &\triangleq [\mathbf{x}_{G,i}(t)]_{i \in G_A}, \\
\mathbf{u}_{G_A}(t) &\triangleq [u_{G,i}(t)]_{i \in G_A}.
\end{aligned} \tag{2}$$

with analogous definitions for $\mathbf{x}_B$, $\mathbf{u}_B$, $\mathbf{x}_{B_A}$, and $\mathbf{u}_{B_A}$. Brackets indexed by a set indicate a stack over that set in ascending index order. Thus $\mathbf{x}_G$ and $\mathbf{x}_B$ have fixed dimensions $3N_G$ and $3N_B$, whereas $\mathbf{x}_{G_A}$ and $\mathbf{x}_{B_A}$ are compact active-roster stacks. The joint propagated vehicle state is $\mathbf{x}(t) = [\mathbf{x}_G(t)^\top, \mathbf{x}_B(t)^\top]^\top$; the active state and control used by a game solve are $\mathbf{x}_A = [\mathbf{x}_{G_A}^\top, \mathbf{x}_{B_A}^\top]^\top$ and $\mathbf{u}_A = [\mathbf{u}_{G_A}^\top, \mathbf{u}_{B_A}^\top]^\top$. Superscripts $0:T$ and $0:T-1$ denote planning-horizon state and control sequences; for example, $\mathbf{x}_{G_A}^{0:T} = (\mathbf{x}_{G_A,0}, \dots, \mathbf{x}_{G_A,T})$ and $\mathbf{u}_{G_A}^{0:T-1} = (\mathbf{u}_{G_A,0}, \dots, \mathbf{u}_{G_A,T-1})$.

The HVAs are fixed at positions collected in $\mathbf{P}_L \in \mathbb{R}^{N_L \times 2}$ and are scenario context rather than dynamic-state components. Every guard receives the common global observation

$$\mathcal{O}_G \triangleq (\mathbf{x}_G, \mathbf{x}_B, \mathbf{P}_L, \mathbf{V}_G, \mathbf{V}_B, \boldsymbol{\sigma}_G, \boldsymbol{\sigma}_B, \boldsymbol{\sigma}_L, t), \tag{3}$$

where $\mathbf{V}_G$ and $\mathbf{V}_B$ are fixed vehicle-speed vectors and the status vectors preserve fixed observation dimensions after entities become inactive. The implemented bandit behavior uses the guard-blind projection $\mathcal{O}_B \triangleq (\mathbf{x}_B, \mathbf{P}_L, \mathbf{V}_B, \boldsymbol{\sigma}_B, \boldsymbol{\sigma}_L, t)$; its actions do not respond to guard state.

1.2 Objectives and Solution Concepts

Guards are modeled as a cooperative team with common cost J_G , while bandits share cost J_B . The detailed terminal-handoff and control-effort costs are specified in Sec. 2.2.2. Let $\gamma_G : \mathcal{O}_G \rightarrow \mathcal{U}_G^{N_G}$ and $\gamma_B : \mathcal{O}_B \rightarrow \mathcal{U}_B^{N_B}$ denote guard and bandit feedback policies—also called “strategies” in game-theory literature [2] and colloquially as “tactics” in this report. A unilateral formulation of the swarm defense problem selects the guard tactics that minimize J_G :

$$\begin{aligned}
\gamma_G^* &= \arg \min_{\gamma_G} J_G(\gamma_G; \gamma_B), \\
\text{such that } \dot{\mathbf{x}} &= f(\mathbf{x}, \gamma_G(\mathcal{O}_G), \gamma_B(\mathcal{O}_B)), \quad \mathbf{x}(0) = \mathbf{x}_0.
\end{aligned} \tag{4}$$

where $f(\mathbf{x}, \gamma_G(\mathcal{O}_G), \gamma_B(\mathcal{O}_B))$ describes the physical dynamics of the drone swarm.

Equation 4 is the unilateral formulation used by the baseline tactics in Sec. 2.1: it optimizes the guard policy for a prescribed bandit policy. The differential-game formulation instead solves for mutually responsive strategies.

$$\begin{aligned}
J_G(\gamma_G^*, \gamma_B^*) &\leq J_G(\gamma_G, \gamma_B^*), \\
J_B(\gamma_G^*, \gamma_B^*) &\leq J_B(\gamma_G^*, \gamma_B),
\end{aligned} \tag{5}$$

Equation 5 defines a *Nash equilibrium*, from which neither team benefits by unilaterally changing its strategy [2, Chapter 6]. The planned bandit strategy is an adversarial model rather than a prediction of the simulated guard-blind bandit behavior. Receding-horizon replanning lets the guard team respond as observed behavior departs from that model.

2 Guard/Defender Tactics

In this section we develop 3-phase tactics used by the team of guard drones; the phases being target assignment, midcourse guidance (MG), and terminal guidance (TG). Fig. 2 provides an illustrative diagram of the control flow between these phases. Generally speaking, the target assignment phase is responsible for assigning to each guard drone a bandit that it is nominally intended to intercept. This phase is generally executed by a centralized planner—mimicking how a command and control node in the real-world would be

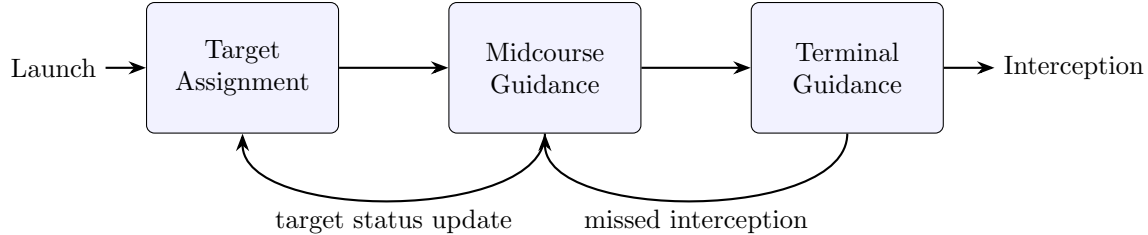


Figure 2: High-level control flow for guard/defender tactics.

responsible for sensing incoming threats and tasking interceptors to meet those threats—and may be run multiple times throughout the engagement, leading to dynamic reassignment of targets (Sec. 2.2.1). The midcourse guidance phase is generally responsible for planning the flyout trajectory of the guards from their current state to a state that best establishes the conditions for an interception. This phase receives much of the attention in the subsequent sections as we develop novel tactics based upon differential game theory (DG) that blend the midcourse guidance and target assignment phases. Terminal guidance is generally responsible for control of a guard drone in the final moments before interception of a bandit. We employ the widely-used, highly-robust technique of proportional navigation (PN) for all of our terminal guidance algorithms; see Sec. 2.2.3 for more discussion of this guidance law.

Not only do we develop the novel game-theoretic tactics (Sec. 2.2) that are the subject of investigation in this report, but also a set of baseline tactics (Sec. 2.1) against which the differential game tactics are compared. All guard tactics, both game-theoretic and baselines, are based upon the same 3-phase architecture given in Fig. 2. They differ in the algorithms that underpin the various phases; most importantly, the midcourse guidance phase.

2.1 Baseline Tactics

To evaluate the effectiveness of the differential game tactics described in Sec. 2.2, it is essential that operationally realistic baseline tactics are developed for comparison. Two baseline tactics are developed: nearest-bandit (NB) and coverage-aware (CA). Both of these tactics are based upon an optimization model of the swarm defense problem—i.e. Eq. 4, in contrast to the game-theoretic model of Eq. (5)—and share the same general solution algorithm, presented in Alg. 1. This algorithm is an instantiation of the 3-phase tactics presented in Fig. 2.

The nearest-bandit (NB) and coverage-aware (CA) tactics differ only in their target assignment phase (Alg. 1, Line 2). As the name implies, the nearest-bandit tactics simply assign to each guard drone the bandit nearest them for interception.

The coverage-aware tactics use a more sophisticated target assignment subroutine that balances between *ease* of interception (e.g. distance from target at assignment) and coverage of all bandits (i.e. incentivizes at least one guard to be assigned to each bandit). This target assignment algorithm is based upon the discounted-redundancy linear assignment optimization described in Sec. 2.2.1.

Algorithm 1 works in an iterative, receding-horizon fashion. At the start of each iteration, a centralized control node checks if there are still active guard and bandit drones. Line 2 represents the target assignment phase previously described. The rest of the algorithm describes how the midcourse guidance (Line 9) and terminal guidance (Line 7) phases are employed in a receding-horizon fashion with updates to the target assignments every T_{replan} seconds (Line 3) or when guards or bandits become inactive due to interceptions (Line 13). The `HeadingErrorGuidance` midcourse guidance subroutine is described in Sec. 2.1.1. The `ExecuteControls` subroutine (Line 12) is where the selected controls are employed and the state of the drone swarms are updated.

Algorithm 1 Swarm Defense Baseline Tactics

```
1: while  $G_A \neq \emptyset$  and  $B_A \neq \emptyset$  do
2:    $(p_{ij}) \leftarrow \text{AssignTargets}(G_A, B_A)$ 
3:   for  $t \leftarrow 0$  to  $T_{\text{replan}}$  do
4:     for all  $i \in G_A$  do
5:        $j \leftarrow \arg \max_{j' \in B_A} p_{ij'}$ 
6:       if  $\text{TerminalHandoffAllowed}(\mathbf{x}_{G,i}, \mathbf{x}_{B,j})$  then
7:          $u_{G,i} \leftarrow \text{ProportionalNavigation}(\mathbf{x}_{G,i}, \mathbf{x}_{B,j})$ 
8:       else
9:          $u_{G,i} \leftarrow \text{HeadingErrorGuidance}(\mathbf{x}_{G,i}, \mathbf{x}_{B,j})$ 
10:      end if
11:    end for
12:     $(\mathbf{x}_G, \mathbf{x}_B, G_A, B_A) \leftarrow \text{ExecuteControls}(\mathbf{u}_G)$ 
13:    if  $G_A$  or  $B_A$  has changed then
14:      break
15:    end if
16:  end for
17: end while
```

2.1.1 Heading Error Midcourse Guidance

With the `HeadingErrorGuidance` controller, the guard seeks to align its heading angle with the line-of-sight (LOS) to the assigned bandit as described in the following equations:

$$\begin{aligned} \psi_{\text{LOS},j/i} &= \text{atan2}(p_{y,B,j} - p_{y,G,i}, p_{x,B,j} - p_{x,G,i}), \\ e_{\psi,i} &= \text{atan2}(\sin(\psi_{\text{LOS},j/i} - \psi_{G,i}), \cos(\psi_{\text{LOS},j/i} - \psi_{G,i})), \\ u_{G,i} &= k_g e_{\psi,i}. \end{aligned} \tag{6}$$

where $u_{G,i}$ is the commanded turn rate of guard i , k_g is a positive heading-error gain, $\psi_{\text{LOS},j/i}$ is the line-of-sight angle from guard i to bandit j , and $e_{\psi,i}$ is the heading error wrapped to $[-\pi, \pi]$. The state-component notation is defined in Sec. 1.1.

2.2 Differential Game Tactics

Algorithm 2 provides pseudocode for implementing the differential game (DG) tactics. Similar to Alg. 1, the DG tactics represent an instantiation of the 3-phase tactics presented in Fig. 2. Algorithm 2 also works in an iterative, receding-horizon fashion. At the start of each iteration, a centralized control node checks if there are still active guard and bandit drones. Line 2 represents the target assignment phase, which for the DG tactics, is actually a soft-assignment; i.e. a *first-guess* at which bandits each guard should pursue. Sec. 2.2.1 describes the `AssignTargets` subroutine for this soft-assignment.

Lines 3–6 represent the game-theoretic components of the midcourse guidance law which are described more thoroughly in Sec. 2.2.2. These components are based upon the Nash equilibrium (Eq. 5) of a linear-quadratic approximation of the swarm defense differential game. The rest of the algorithm describes how the differential game strategy is employed in a receding-horizon fashion with updates to the target soft-assignments every T_{replan} seconds (Line 7) or when guards or bandits become inactive due to interceptions (Line 16). Lines 9 and 10 represent the terminal guidance handoff to the proportional navigation controller described in Sec. 2.2.3.

2.2.1 Dynamic Target Soft-Assignment

In the `AssignTargets` subroutine of Alg. 2 (Line 2), a shared coordinator, akin to a centralized command and control node, provides *soft-assignments* between active guards and incoming bandits to be intercepted. The assignment is “soft” in the sense that it is only an initial guess as to which bandit a guard will intercept; however, the midcourse guidance phase (Sec. 2.2.2) may ultimately result in a different interception.

Algorithm 2 Swarm Defense Differential Game Tactics

```
1: while  $G_A \neq \emptyset$  and  $B_A \neq \emptyset$  do
2:    $(p_{ij}) \leftarrow \text{AssignTargets}(G_A, B_A)$ 
3:    $(\bar{\mathbf{x}}_A^{0:T}, \bar{\mathbf{u}}_A^{0:T-1}) \leftarrow \text{FormReferenceTrajectories}((p_{ij}), \mathbf{x}_{G_A}, \mathbf{x}_{B_A})$ 
4:    $(J_G, J_B) \leftarrow \text{DefineGameCosts}(G_A, B_A)$ 
5:    $lqgame \leftarrow \text{ApproxLQGame}(\bar{\mathbf{x}}_A^{0:T}, \bar{\mathbf{u}}_A^{0:T-1}, J_G, J_B)$ 
6:    $\gamma_G \leftarrow \text{SolveFeedbackNash}(lqgame)$ 
7:   for  $t \leftarrow 0$  to  $T_{\text{replan}}$  do
8:     for all  $i \in G_A$  do
9:       if  $\text{TerminalHandoffAllowed}(i, \mathbf{x}_G, \mathbf{x}_B)$  then
10:         $u_{G,i} \leftarrow \text{ProportionalNavigation}(i, \mathbf{x}_G, \mathbf{x}_B)$ 
11:       else
12:         $u_{G,i} \leftarrow \gamma_{G,i}(t, \mathbf{x}_{G_A}, \mathbf{x}_{B_A})$ 
13:       end if
14:     end for
15:      $(\mathbf{x}_G, \mathbf{x}_B, G_A, B_A) \leftarrow \text{ExecuteControls}(\mathbf{u}_G)$ 
16:     if  $G_A$  or  $B_A$  has changed then
17:       break
18:     end if
19:   end for
20: end while
```

Assigning to each guard a bandit to be pursued requires an algorithm that accounts for the relative *difficulty* of all possible guard-bandit pairings and balances this against the desire to intercept each and every bandit, with the constraint that each guard may only target one bandit at a time. In order to handle all cases of relative guard/bandit swarm sizes, the soft-assignment algorithm must be—in general—a many-to-one, potentially non-surjective function; i.e. multiple elements of the guard set may target the same bandit in the bandit set, and some bandits may have no guard assigned to them.

Using the total and active-roster notation of Sec. 1.1, let $p_{ij} \in \{0, 1\}$ indicate whether guard $i \in G_A$ is assigned to bandit $j \in B_A$. Let the number of guards assigned to bandit j be

$$P_j = \sum_{i \in G_A} p_{ij}. \quad (7)$$

To model the assignment problem, we assume that there is both a *cost*, $c_{ij}^{\text{TA}} \geq 0$, associated with pairing p_{ij} , as well as a reward, $v_j \geq 0$, for having had *some* guard assigned to bandit j (i.e. the assignment reward is agnostic to which guard is assigned). We then apply a redundancy discount factor, $\beta \in [0, 1]$, that reduces the marginal reward of assigning more than one guard to a bandit. Therefore, we have the total payoff of assignments to bandit j given as

$$V_j(P_j) = v_j \sum_{k=0}^{P_j-1} \beta^k, \quad V_j(0) = 0. \quad (8)$$

Thus, the first assigned guard contributes reward v_j , the second adds βv_j , and each additional guard has weakly smaller marginal reward.

Therefore, we can define the target (soft) assignment problem as

$$\max_{p_{ij}} \quad \sum_{j \in B_A} V_j(P_j) - \sum_{i \in G_A} \sum_{j \in B_A} c_{ij}^{\text{TA}} p_{ij}, \quad (9)$$

$$\text{subject to} \quad \sum_{j \in B_A} p_{ij} = 1, \quad \forall i \in G_A, \quad (10)$$

$$p_{ij} \in \{0, 1\}, \quad \forall i \in G_A, j \in B_A. \quad (11)$$

Each guard must be assigned to one-and-only-one bandit, while a bandit may receive zero, one, or multiple guards. The case $\beta = 0$ values only first coverage of each bandit; values $0 < \beta < 1$ reward redundancy with diminishing returns; and $\beta = 1$ assigns full value to every additional guard.

In our experiments, the cost, bandit reward, and redundancy discount are

$$c_{ij}^{\text{TA}} = w_{\text{TA}} \frac{\|\mathbf{p}_{B,j} - \mathbf{p}_{G,i}\|}{\max_{i' \in G_A, j' \in B_A} \|\mathbf{p}_{B,j'} - \mathbf{p}_{G,i'}\|}, \quad v_j = 1, \quad \beta = 0.1, \quad w_{\text{TA}} = 0.25, \quad (12)$$

where $c_{ij}^{\text{TA}} = 0$ when the normalizing maximum is zero.

The discounted-redundancy objective is solved by expanding each active bandit into N_{G_A} virtual assignment slots with successively discounted marginal rewards, then applying a rectangular linear-assignment solver. The selected slot for each guard maps directly to its associated physical bandit. The full linear reformulation is given in Appendix A.1.

2.2.2 Game Theoretic Midcourse Guidance

The differential-game (DG) midcourse guidance in lines 3–6 of Alg. 2 repeatedly constructs and solves a two-player, finite-horizon approximation of the swarm-defense game whose high-level definition is given in Eq. (5). The bandit player is an adversarial planning model, rather than a prediction of the simulated bandits' guard-blind behavior (Sec. 1.1).

At each replan, (p_{ij}) supplies one provisional bandit target for each active guard. These assignments seed the nominal active-roster trajectories $(\bar{\mathbf{x}}_A^{0:T}, \bar{\mathbf{u}}_A^{0:T-1})$: guards use heading-error guidance toward their provisionally assigned bandits, while bandits use zero nominal turn rate. Propagation under the nonlinear Dubins dynamics produces a dynamically feasible operating trajectory. The provisional assignment is used only for this initialization; the terminal objective—later summarized in Eq. (14)—evaluates every pair in $G_A \times B_A$ and can therefore favor a different pairing.

At its core, the DG guidance law encodes the guard team's desire for optimal conditions *at the end of midcourse guidance* to handoff to terminal guidance. More specifically, guards favor terminal-handoff pairs that are close in distance and require little PN maneuvering (Sec. 2.2.3). Therefore, at the terminal handoff time t_f , each active guard–bandit pair is evaluated by the differentiable terminal-handoff kernel c_{ij}^{TH} . Let $\mathbf{r}_{j/i} = \mathbf{p}_{B,j} - \mathbf{p}_{G,i}$ and $\mathbf{v}_{j/i} = \mathbf{v}_{B,j} - \mathbf{v}_{G,i}$ denote the relative position and velocity from guard i to bandit j .

$$\begin{aligned} R_{\epsilon,ij} &= \sqrt{\mathbf{r}_{j/i}^\top \mathbf{r}_{j/i} + \epsilon_R^2}, \\ d_{\epsilon,ij} &= R_{\epsilon,ij} - \epsilon_R, \\ V_{c,ij} &= -\frac{\mathbf{r}_{j/i}^\top \mathbf{v}_{j/i}}{R_{\epsilon,ij}}, \\ \dot{\lambda}_{ij} &= \frac{r_{x,j/i} v_{y,j/i} - r_{y,j/i} v_{x,j/i}}{R_{\epsilon,ij}^2}, \\ a_{\text{PN},ij} &= \alpha_G V_{c,ij} \dot{\lambda}_{ij}, \\ c_{ij}^{\text{TH}} &= w_{gb} d_{\epsilon,ij} + w_{\text{PN}} \left(\sqrt{a_{\text{PN},ij}^2 + \epsilon_a^2} - \epsilon_a \right). \end{aligned} \quad (13)$$

Here α_G is the guard PN navigation constant (typically set to 3.0; see Sec. 2.2.3) and $V_{c,ij} > 0$ denotes a closing engagement. The two positive regularization parameters make the range and acceleration-magnitude terms differentiable at zero range and zero lateral acceleration; away from these geometries, they approach $\|\mathbf{r}_{j/i}\|$ and $|a_{\text{PN},ij}|$, respectively.

The guard and bandit swarms are each controlled by one team coordinator. Rather than fixing the provisional assignment in the terminal objective, the guard team takes a differentiable soft minimum (i.e. LogSumExp) over every active bandit for each active guard:

$$g_{f,G}^{\text{TH}} = \begin{cases} \sum_{i \in G_A} \left[-\tau \log \left(\sum_{j \in B_A} \exp \left(-\frac{c_{ij}^{\text{TH}}}{\tau} \right) \right) \right], & B_A \neq \emptyset, \\ 0, & B_A = \emptyset, \end{cases} \quad (14)$$

where $\tau > 0$ is the soft-assignment temperature. As τ approaches zero, the LogSumExp soft minimum approaches the hard minimum, but becomes less numerically well conditioned under differentiation. The terminal bandit objective is its negative, $g_{f,B}^{\text{TH}} = -g_{f,G}^{\text{TH}}$.

Turn-rate limits are represented in the game by normalized quadratic effort penalties:

$$\begin{aligned} g_{t,G}(\mathbf{u}_{G_A}) &= w_{u,G} \sum_{i \in G_A} \left(\frac{u_{G,i}}{\bar{\omega}_G} \right)^2, \\ g_{t,B}(\mathbf{u}_{B_A}) &= w_{u,B} \sum_{j \in B_A} \left(\frac{u_{B,j}}{\bar{\omega}_B} \right)^2. \end{aligned} \quad (15)$$

where $\bar{\omega}_G$ and $\bar{\omega}_B$ are the maximum turn rates and $w_{u,G}$ and $w_{u,B}$ are effort weights. The complete team costs are

$$\begin{aligned} J_G &= g_{f,G}^{\text{TH}}(\mathbf{x}_A(t_f)) + \int_{t_0}^{t_f} g_{t,G}(\mathbf{u}_{G_A}(t)) dt, \\ J_B &= g_{f,B}^{\text{TH}}(\mathbf{x}_A(t_f)) + \int_{t_0}^{t_f} g_{t,B}(\mathbf{u}_{B_A}(t)) dt, \quad g_{f,B}^{\text{TH}} = -g_{f,G}^{\text{TH}}. \end{aligned} \quad (16)$$

The nonlinear dynamics and costs in Eqs. (1) and (16) are locally approximated about the nominal trajectory. With $\delta \mathbf{x}_k = \mathbf{x}_{A,k} - \bar{\mathbf{x}}_{A,k}$ and $\delta \mathbf{u}_k = \mathbf{u}_{A,k} - \bar{\mathbf{u}}_{A,k}$, the discrete-time dynamics and each team cost are approximated as

$$\begin{aligned} \delta \mathbf{x}_{k+1} &\approx A_k \delta \mathbf{x}_k + B_k \delta \mathbf{u}_k, \\ J_q &\approx J_q(\bar{\mathbf{x}}_A^{0:T}, \bar{\mathbf{u}}_A^{0:T-1}) + \mathbf{h}_q^\top \delta \mathbf{z} + \frac{1}{2} \delta \mathbf{z}^\top H_q \delta \mathbf{z}, \quad q \in \{G, B\}, \end{aligned} \quad (17)$$

where A_k and B_k are the dynamics Jacobians and $\delta \mathbf{z}$ stacks the state and control deviations over the planning horizon. The gradients and Hessians defining $\mathbf{h}_q$ and H_q are obtained by automatic differentiation. These linear dynamics and quadratic costs form an LQ game, whose feedback Nash equilibrium is computed by Riccati recursion [2, 3] to obtain the guard policy γ_G . In practice, the PYDGENS library [1] is used for both the LQ approximation (ApproxLQGame) and Riccati recursion (SolveFeedbackNash).

The guards execute γ_G during midcourse guidance until a guard transitions to PN terminal guidance. Only a short prefix of the planned policy is used: the cycle repeats after T_{replan} or whenever G_A or B_A changes, rebuilding the assignment, nominal trajectory, and local game from the new state.

2.2.3 Terminal Guidance

Proportional navigation (PN) is a terminal homing law that commands lateral acceleration in proportion to the line-of-sight (LOS) angular rate. It is widely used because it is comparatively simple while directly responding to the relative geometry that determines miss distance [6].

For guard $i \in G_A$ and bandit $j \in B_A$, use the relative position and velocity introduced for the terminal-handoff cost in Eq. (13): $\mathbf{r}_{j/i} = \mathbf{p}_{B,j} - \mathbf{p}_{G,i}$ and $\mathbf{v}_{j/i} = \mathbf{v}_{B,j} - \mathbf{v}_{G,i}$. Let $R_{ij} = \|\mathbf{r}_{j/i}\|$ and $\lambda_{ij} = \text{atan2}(r_{y,j/i}, r_{x,j/i})$. The planar LOS rate and closing speed are

$$\dot{\lambda}_{ij} = \frac{r_{x,j/i} v_{y,j/i} - r_{y,j/i} v_{x,j/i}}{R_{ij}^2}, \quad V_{c,ij} = -\frac{\mathbf{r}_{j/i}^\top \mathbf{v}_{j/i}}{R_{ij}}, \quad (18)$$

where $V_{c,ij} > 0$ denotes decreasing range. The scalar PN normal-acceleration command for guard G_i against bandit B_j is

$$a_{\text{PN},ij} = \alpha_G V_{c,ij} \dot{\lambda}_{ij}, \quad (19)$$

where $\alpha_G > 0$ is the guard's dimensionless navigation constant. The sign of $a_{\text{PN},ij}$ selects the turn direction. For a fixed-speed planar Dubins guard with speed V_G , this lateral-acceleration command corresponds to the heading-rate command $\omega_{\text{PN},ij} = a_{\text{PN},ij}/V_G$.

The scalar law in Eq. (19) does not by itself specify the direction in which guard G_i realizes the commanded acceleration. In *true PN*, the commanded acceleration is normal to the LOS; in *pure PN*, it is

normal to the guard velocity (equivalently, normal to its body axis in this planar fixed-speed model). The terminal guard policy implements pure PN by converting the scalar command to a Dubins turn rate. The differential-game cost in Eq. (13) uses the magnitude of that same scalar command only as a measure of terminal-handoff maneuvering demand; it does not propagate a PN controller during the midcourse game solve.

3 Simulation Environment

The guard tactics (Sec. 2) are evaluated within a simulation environment built in Python using the PettingZoo API to model multi-agent decision processes [8]. The linear-quadratic game in the DG tactics (Alg. 2, Line 6) is solved with the PYDGENS library [1]. The system dynamics (Eq. (1)) are integrated over fixed time steps, typically of 1.0 seconds. The information and action structure of the simulated teams is defined in Sec. 1.1; the remainder of this section describes its implementation and engagement-specific assumptions. Section 3.1 describes how interceptions of bandits and assets are adjudicated. In contrast to the guard team which is modeled as an *agent*—i.e. whose tactics are separable from the simulation environment—the bandit team is modeled as a *bot* whose tactics are part of the environment. Section 3.2 describes two different scenarios for the simulation based upon differing bandit behaviors: non-evasive bandits (B1) and evasive bandits (B2). Section 4 then describes how the simulation environment is used in a set of Monte Carlo experiments, along with the performance metrics and Bayesian statistics used for evaluation of tactics performance.

3.1 Interception Adjudication

The determination of whether a simulated guard intercepts a bandit and whether a simulated bandit reaches a HVA is key to the evaluation of the “success” of either team. When a guard successfully intercepts a bandit, both vehicles are eliminated; the bandit then poses no threat to assets and the guard is not able to intercept any further bandits. However, due to large simulation time steps (e.g. 1.0 seconds) used to accelerate computation in large Monte Carlo campaigns, we cannot simply implement a narrow interception range as it would often be passed through, unrecorded, from one time step to the next. Instead, we use a dynamically-scaled *effective* interception range (Eq. (20)) that increases in size when guard and bandit headings are aligned with a small relative velocity. This tends to bias the interception adjudication toward approach-from-behind configurations, which is a common technique for real-world interceptors.

For each active guard–bandit pair $(i, j) \in G_A \times B_A$, reuse the relative position $\mathbf{r}_{j/i}$ and velocity $\mathbf{v}_{j/i}$ defined for the terminal-handoff kernel in Eq. (13). Let $R_{ij} = \|\mathbf{r}_{j/i}\|$ and $\hat{\ell}_{j/i} = \mathbf{r}_{j/i}/R_{ij}$ denote the corresponding range and guard-to-bandit line-of-sight unit vector. The environment declares an interception when

$$\text{intercept}_{ij} \iff R_{ij} \leq R_{\text{eff},ij},$$

$$R_{\text{eff},ij} = R_{\text{gb}} + \underbrace{(R_{\text{adj}} - R_{\text{gb}}) \text{clip}_{[0,1]} \left(\frac{-\hat{\ell}_{j/i}^T \mathbf{v}_{j/i}}{\|\mathbf{v}_{j/i}\|} \right)}_{\text{closing alignment}} \underbrace{\text{clip}_{[0,1]} \left(1 - \frac{\|\mathbf{v}_{j/i}\|}{V_{\text{rel}}} \right)}_{\text{relative-speed factor}}. \quad (20)$$

Here R_{gb} is the physical guard–bandit interception radius, R_{adj} is the maximum adjudication range, and V_{rel} is the relative-speed scale. Thus, the envelope expands beyond the physical radius only for a closing, nearly collinear, low-relative-speed encounter; it reduces to R_{gb} for non-closing geometry or relative speed at or above V_{rel} . The implementation assigns zero closing alignment when either range or relative speed is zero.

3.2 Bandit/Intruder Tactics

The swarm of bandit drones also employs 3-phase tactics, as depicted in Fig. 2, with target assignment, midcourse guidance, and terminal guidance phases. The bandit tactics differ from the baseline guard tactics (Sec. 2.1) in that they employ static target assignment (i.e. HVA targets are assigned to each bandit at launch and do not change throughout the engagement), and that they cannot observe—and, therefore, are not reactive to—the state of the guard drones (see Sec. 1.1).

Two different behavior models for the bandits are studied. First is that of *non-evasive* midcourse guidance—labeled simply as Behavior 1 or B1—whereby bandits navigate to their assigned HVA targets using heading-error midcourse guidance described in Sec. 2.1.1. The second behavior—labeled Behavior 2, B2—uses sporadic *evasive* maneuvers during midcourse guidance.

In the B2 version of the simulation environment, each bandit receives an immutable, open-loop “jink” schedule at initialization. The schedule is constructed based upon the bandit’s initial time-to-go to its fixed target, assuming a straight-line trajectory; jinks are placed in non-overlapping intervals spanning the middle portion of that nominal engagement. For each scheduled jink, the start time and turn direction are sampled, while the turn-rate magnitude and jink duration are pre-specified parameters of the environment. During the jink interval, the nominal heading-error command is replaced with the sampled turn-rate command, temporarily deviating the bandit from its nominal target track. At the end of the interval, the bandit resumes the heading-error midcourse guidance control described in Sec. 2.1.1.

It is worth noting that, in spite of the use of game-theoretic techniques described in Sec. 2.2 that compute an equilibrium strategy for the bandit swarm, this strategy is not actually in use by the bandits.

4 Experiment Design

In order to evaluate the effectiveness of the differential game (DG) tactics (Sec. 2.2) compared to baseline tactics (Sec. 2.1), we use the simulation environment (Sec. 3) to conduct a Monte Carlo campaign of randomized trials, measure performance metrics, and perform Bayesian analysis of the performance metrics to estimate the probability that differential game tactics outperform the baselines.

4.1 Monte Carlo Trials

Two separate Monte Carlo campaigns are conducted: one with the non-evasive bandit behaviors (B1) and one with evasive bandits (B2). Table 1 summarizes the parameters that are randomized between Monte Carlo trials. Unless otherwise indicated, numerical intervals in Table 1 are uniform sampling ranges; initial positions are sampled independently and uniformly within their stated rectangles. The B2 sampler shares every B1 parameter and adds the open-loop jink (i.e. evasive maneuver) parameters marked with an asterisk.

The same random number seed for randomizing trial parameters is used across the coverage-aware (CA) and differential game (DG) experiment campaigns. Therefore, the CA and DG tactics are evaluated with trials of identical parameters. This enables a more direct comparison of Bayesian posterior distributions of performance metrics; discussed in the next section.

4.2 Performance Metrics & Bayesian Analysis

The effectiveness of guard tactics is evaluated based upon performance metrics described in Table 2. These attempt to measure how effectively the guards have defended the high-value assets (HVAs or “ladies” in Rusnak’s terminology [7]) and how much control effort was required on the part of the guards.

The boolean *defense-success* outcome is of primary concern for our performance analysis. For a given Monte Carlo trial episode, defense-success Y_{sd} is True if and only if all bandits are intercepted and no HVAs are reached. For statistical analysis, a Beta-Binomial model [4, Ch. 2] is applied whereby defense-success is treated as a Bernoulli trial and we seek to estimate the probability of successful defense for a particular set of guard tactics. We use a Beta distribution to model the success probability parameter, θ_{sd} , and a noninformative (uniform) prior on the distribution.

The *HVAs-reached* ($S_{lr} \in [0, 1]$) and *bandits-intercepted* ($S_{bi} \in [0, 1]$) trial outcomes are real-valued scalars that measure, respectively, the percent of assets reached by bandits and percent of bandits intercepted by guards within a simulated episode. For statistical analysis, a Dirichlet-Multinomial model [4, Ch. 3] is applied to these outcomes whereby the real-valued outcomes, S_{lr} and S_{bi} , are binned into K -bins evenly distributed in range $[0, 1]$. Then Monte Carlo trial data is used to estimate the posterior distribution of the probability mass across the bins. A weak, symmetric Dirichlet prior is used.

Note that, while defense-success and bandits-intercepted outcomes are related, there are salient differences between the two. Beyond the fact that one is a boolean while the other is a real-valued scalar, you can get a very different perception of guard tactics effectiveness from these two values. For example, if there were

Table 1: Monte Carlo trial parameters for non-evasive (B1) and evasive (B2) bandit scenarios. Parameters marked with * apply only to the B2 evasive-bandit scenario.

Category	Parameter	Value or sampling range	Scenario
Campaign	Integration time step, maximum steps	1 s, 500 (500 s maximum duration)	B1, B2
Team size	Assets N_L	UnifInt{4, ..., 8}	B1, B2
	Bandits N_B	UnifInt{8, ..., 12}	B1, B2
	Guards N_G	UnifInt{max(10, N_B), ..., 16}	B1, B2
Vehicle	Guard speed	[70, 80] m/s	B1, B2
	Bandit speed	[45, 55] m/s	B1, B2
	Guard maximum turn rate	[0.8, 1.2] rad/s	B1, B2
	Bandit maximum turn rate	[0.45, 0.55] rad/s	B1, B2
Adjudication	Guard–bandit physical radius	[2.5, 3.5] m	B1, B2
	Guard–bandit maximum adjudication range	[95, 105] m	B1, B2
	Guard–bandit relative-speed scale	[95, 105] m/s	B1, B2
	Bandit–asset physical radius	[9.9, 10.1] m	B1, B2
	Bandit–asset adjudication range	[95, 105] m	B1, B2
Initial state	Guard and asset position	$x, y \in [-5,000, 5,000]$ m	B1, B2
	Bandit position	$x \in [-20,000, 20,000]$ m; $y \in [10,000, 15,000]$ m	B1, B2
	Initial-heading noise half-width	[0.2, 0.3] rad	B1, B2
Bandit guidance	PN navigation constant	[2.5, 3.5]	B1, B2
	Midcourse heading-error gain	[0.4, 0.6] s ⁻¹	B1, B2
	Terminal-handoff range	[80, 120] m	B1, B2
	Terminal LOS-error threshold	$[\pi/3, 2\pi/3]$ rad	B1, B2
Open-loop jink	Number of jinks*	UnifInt{1, ..., 3}	B2*
	Jink duration*	[3, 8] s	B2*
	Jink turn-rate fraction*	[0.5, 0.9] of bandit maximum turn rate	B2*

a set of guard tactics that *always* intercepted 99 out of 100 bandits within an engagement, then you would have bandit-intercepted rate of 99% but a defense-success rate of 0% because in no trial do you ever catch all of the bandits.

The *mean-control-saturation* ($S_{\text{sat}} \in [0, 1]$) trial outcome measures the proportion of time steps in which any guard agent’s control is saturated (i.e. the control command exceeds the actuator limits of the guard vehicle and must be clipped to the bounds), averaged across all guard agents using the same tactics. A Dirichlet-Multinomial with a weak, symmetric prior is similarly applied to estimate the posterior distribution of mean-control-saturation.

As previously mentioned, the differential game (DG) and coverage-aware (CA) experiment campaigns use identical ordered trial realizations, i.e. the same random number seed. We can use this information to more directly answer questions such as: *what is the probability that the differential game tactics have a higher defense-success probability than coverage-aware tactics?* To do so, we define the four possible defense-success outcomes of a paired trial: both DG and CA are successful (1,1), only DG succeeds (1,0), only CA succeeds (0,1), and both fail (0,0). By counting the paired outcomes across DG and CA campaigns, we form the Dirichlet posterior distribution as

$$(\pi_{11}, \pi_{10}, \pi_{01}, \pi_{00}) \mid D \sim \text{Dirichlet}(n_{11} + \frac{1}{2}, n_{10} + \frac{1}{2}, n_{01} + \frac{1}{2}, n_{00} + \frac{1}{2}) \quad (21)$$

where n_{ab} is the paired-trial count and π_{ab} is the joint probability of DG outcome a and CA outcome b . By recognizing that $\theta_{\text{sd}}^{\text{DG}} = \pi_{11} + \pi_{10}$ and $\theta_{\text{sd}}^{\text{CA}} = \pi_{11} + \pi_{01}$, we can sample the distribution in Eq. 21 to estimate $\Pr(\theta_{\text{sd}}^{\text{DG}} > \theta_{\text{sd}}^{\text{CA}} \mid D^{\text{DG}}, D^{\text{CA}})$.

Table 2: Per-trial performance metrics and Bayesian analyses. The reported posterior intervals are central 95% credible intervals.

Metric	Per-trial observation	Bayesian model and prior	Posterior quantities reported
defense-success	$Y_{sd} \in \{0, 1\}$, equal to one only if no asset is reached and every bandit is intercepted; zero otherwise	Bernoulli likelihood with $\theta_{sd} \sim \text{Beta}(1, 1)$	$\mathbb{E}[\theta_{sd} D]$ and its 95% credible interval estimated from 100,000 posterior samples; higher is better
HVAs-reached	$S_{lr} \in [0, 1]$, the fraction of assets reached in one trial	One hundred equal-width bins on $[0, 1]$; multinomial likelihood with symmetric Dirichlet($1/K, \dots, 1/K$) prior for $K = 100$ bins (total concentration one)	$\mathbb{E}[\mu_{lr} D]$ and its 95% credible interval estimated from 100,000 posterior samples; lower is better
Bandits intercepted	$S_{bi} \in [0, 1]$, the fraction of bandits intercepted in one trial	One hundred equal-width bins on $[0, 1]$; multinomial likelihood with symmetric Dirichlet($1/K, \dots, 1/K$) prior for $K = 100$ bins (total concentration one)	$\mathbb{E}[\mu_{bi} D]$ and its 95% credible interval estimated from 100,000 posterior samples; higher is better
mean-control-saturation	$S_{sat} \in [0, 1]$, the mean across guards of the fraction of recorded active control steps on which the requested action is clipped	One hundred equal-width bins on $[0, 1]$; multinomial likelihood with symmetric Dirichlet($1/K, \dots, 1/K$) prior for $K = 100$ bins (total concentration one)	$\mathbb{E}[\mu_{sat} D]$ and its 95% credible interval estimated from 100,000 posterior samples; lower is better

5 Results & Discussion

Tables 3 and 4 report the Bayesian posteriors of performance metrics (Table 2) for the non-evasive (B1) and evasive (B2) intruder environments, respectively. Generally speaking, we seek high expected values for defense-success and bandits-intercepted (θ_{sd} and μ_{bi}) and low expected values for HVAs-reached and control saturation (μ_{lr} and μ_{sat}).

The key result from Tables 3 and 4 is that the differential game tactics produce the highest probability of successful defense of high-value assets ($\mathbb{E}[\theta_{sd} | D]$); higher than both the naive nearest-bandit (NB) and more sophisticated coverage-aware (CA) baseline tactics that ignore the game-theoretic modeling of the problem, instead treating it as a unilateral optimization (Eq. (4)). The importance of the game modeling is further emphasized when comparing results between Tables 3, where the bandits were non-evasive, and 4, where bandits performed evasive maneuvers: we see a modest drop in successful defense when bandits employ evasive maneuvers for the non-game-theoretic baseline tactics (NB and CA); whereas the differential game (DG) tactics are able to maintain their performance.

Tables 3 and 4 also help illustrate the importance of intercepting *all* bandits within an engagement, versus simply intercepting a large majority per engagement. In the case of evasive bandits, we see that the nearest-bandit (NB) guard tactics are capable of intercepting an average of 80.2% of bandits; however, this only results in a 14.6% percent chance that *all* bandits are intercepted during a particular engagement. In the real world, it would be unacceptable to have such a low chance of defending all high-value assets. We see how the coverage-aware (CA) guard tactics significantly improve upon this key metric, raising the probability of successfully defending *all* HVAs to 94.6%. However, the differential game (DG) tactics further improve on this key metric, raising the defense success probability to 96.8% in the presence of evasive bandits. Another way to frame this is that the non-game-theoretic CA tactics have a 5.4% *gap to perfect defense*, and by employing game-theoretic tactics, 40.7% of this gap may be closed.

Table 3: Bayesian performance summary for **non-evasive bandit tactics**. Each entry gives the posterior mean and central 95% credible interval (in brackets). The nearest-bandit baseline uses 1,000 Monte Carlo trials; the coverage-aware baseline and differential-game tactics each use 500. Posterior summaries use 100,000 samples.

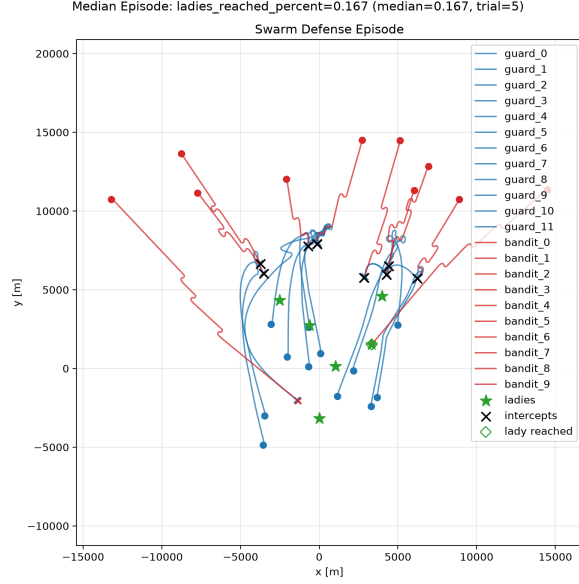
Guard tactic	Defense success $\mathbb{E}[\theta_{\text{sd}} \mid D]$	HVAs reached $\mathbb{E}[\mu_{\text{lr}} \mid D]$	Bandits intercepted $\mathbb{E}[\mu_{\text{bi}} \mid D]$	Control saturation $\mathbb{E}[\mu_{\text{sat}} \mid D]$
Nearest-bandit (NB)	0.157 [0.135, 0.180]	0.182 [0.172, 0.194]	0.828 [0.821, 0.835]	0.007 [0.007, 0.009]
Coverage-aware (CA)	0.958 [0.939, 0.974]	0.013 [0.010, 0.018]	0.990 [0.987, 0.992]	0.008 [0.007, 0.011]
Differential-game (DG)	0.966 [0.949, 0.980]	0.011 [0.008, 0.016]	0.991 [0.988, 0.993]	0.006 [0.005, 0.009]
<i>Paired DG–coverage-aware comparison (500 trials): $\Pr(\theta_{\text{sd}}^{\text{DG}} > \theta_{\text{sd}}^{\text{CA}} \mid D^{\text{DG}}, D^{\text{CA}}) = 0.858$</i>				

Table 4: Bayesian performance summary for **evasive bandit tactics**. Each entry gives the posterior mean and central 95% credible interval (in brackets). The nearest-bandit baseline uses 1,000 Monte Carlo trials; the coverage-aware baseline and differential-game tactics each use 500. Posterior summaries use 100,000 samples.

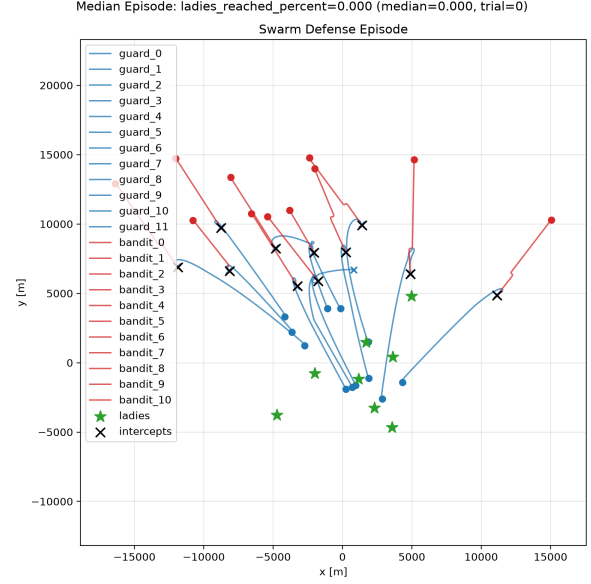
Guard tactic	Defense success $\mathbb{E}[\theta_{\text{sd}} \mid D]$	HVAs reached $\mathbb{E}[\mu_{\text{lr}} \mid D]$	Bandits intercepted $\mathbb{E}[\mu_{\text{bi}} \mid D]$	Control saturation $\mathbb{E}[\mu_{\text{sat}} \mid D]$
Nearest-bandit (NB)	0.146 [0.125, 0.168]	0.192 [0.181, 0.203]	0.802 [0.794, 0.810]	0.009 [0.008, 0.010]
Coverage-aware (CA)	0.946 [0.925, 0.964]	0.015 [0.011, 0.019]	0.990 [0.986, 0.992]	0.009 [0.008, 0.013]
Differential-game (DG)	0.968 [0.951, 0.982]	0.011 [0.009, 0.016]	0.991 [0.988, 0.993]	0.006 [0.005, 0.009]
<i>Paired DG–coverage-aware comparison (500 trials): $\Pr(\theta_{\text{sd}}^{\text{DG}} > \theta_{\text{sd}}^{\text{CA}} \mid D^{\text{DG}}, D^{\text{CA}}) = 0.999$</i>				

Given the relatively high success rate of both the DG and CA guard tactics, it may be asked if differential game tactics are probabilistically “better” than the non-game-theoretic baselines, or if these differences may be statistically insignificant. To answer this question, Tables 3 and 4 also provide results for the paired-trial posterior analysis described in Sec. 4.2 and show that the probability that DG tactics have a higher probability of successful defense than CA tactics is 85.8% for non-evasive bandits, rising to 99.9% for evasive bandits.

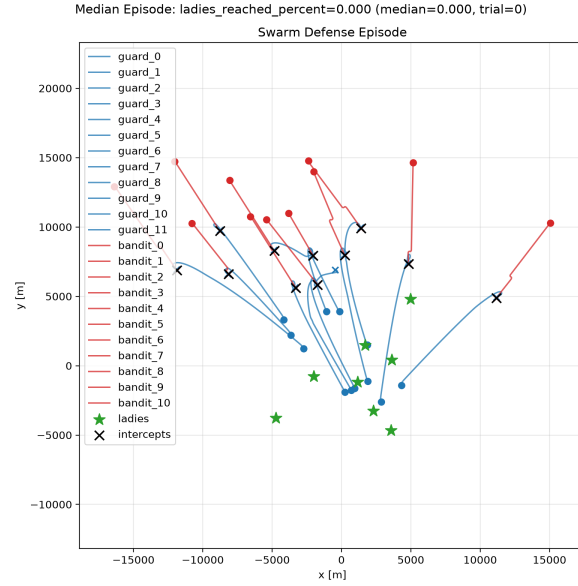
Figures 3 and 4 illustrate the vehicle trajectories for a selection of evasive-bandit (B2) episodes with the nearest-bandit (NB), coverage-aware (CA), and differential-game (DG) guard tactics. Figure 3 represents a median trial in each of the three campaigns as measured by the percentage of HVAs-reached by bandits; Figure 4 represents the worst trial from each campaign. These figures are presented, mostly, to give intuition of what an episode rollout would look like and how guards distribute themselves across bandits. It is difficult to discern meaningful differences in the behavior of the CA and DG tactics within these individual trials, which is why the statistical results in Tables 3 and 4 are the more salient for our analysis.



(a) Nearest-bandit baseline

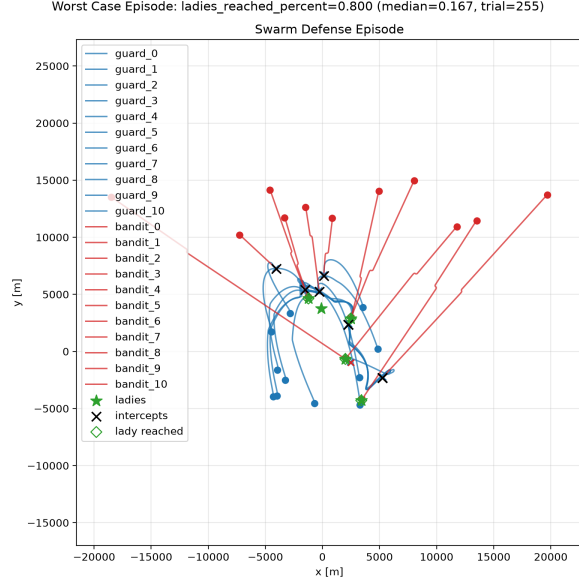


(b) Coverage-aware baseline

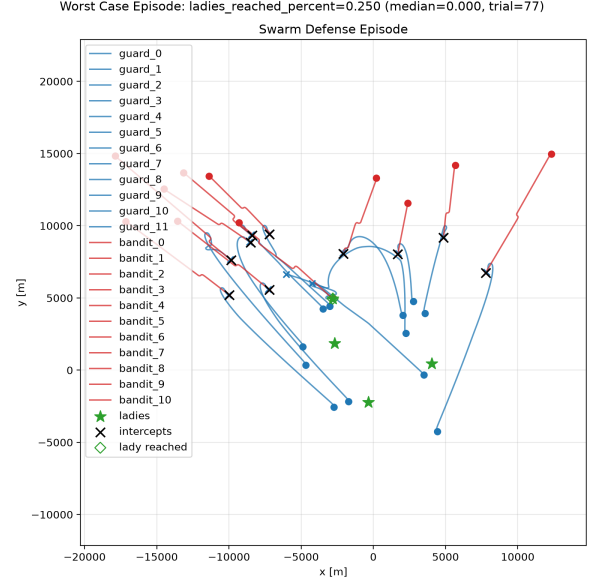


(c) Differential-game tactics

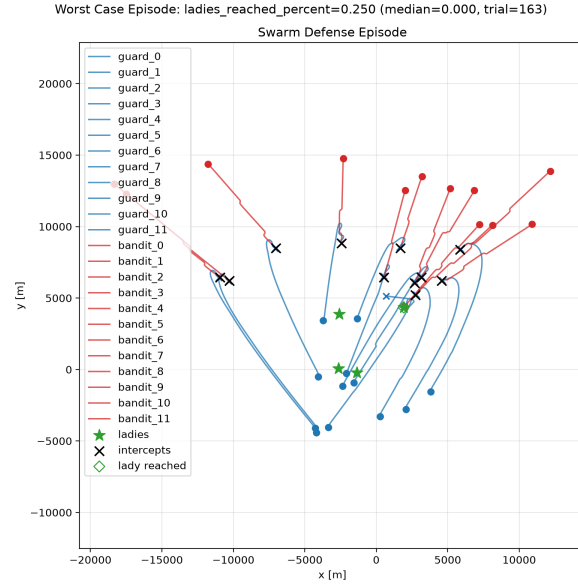
Figure 3: Median evasive-bandit (B2) episode selected independently for each tactic by HVA-reached percentage: (a) nearest-bandit baseline (0.167), (b) coverage-aware baseline (0.000), and (c) differential-game tactics (0.000).



(a) Nearest-bandit baseline



(b) Coverage-aware baseline



(c) Differential-game tactics

Figure 4: Worst observed evasive-bandit (B2) episode selected independently for each tactic by HVA-reached percentage: (a) nearest-bandit baseline (0.800), (b) coverage-aware baseline (0.250), and (c) differential-game tactics (0.250).

6 Limitations & Future Work

There are a number of limitations to this current work that need addressing in future work in order to mature game-theoretic algorithms for use in real-world swarm defense. It has been assumed that all guard drones—or at least a centralized guard coordinator—may observe the state of all drones and assets at all times. It has been assumed that a centralized guard coordinator may communicate with all guard drones at all times, with no bandwidth limitations. A highly simplified aircraft model has been employed; neglecting even variations in altitude, let alone high-fidelity aerodynamics and flight control. The interception adjudication model is also highly simplified. Sensor models, such as cameras and accelerometers, would be needed to improve simulation of interception dynamics.

While we have modeled both non-evasive and evasive (i.e. pre-scripted, open-loop) bandits, we have not yet simulated *truly reactive* bandits which are able to perceive the state of guard drones and maneuver accordingly to better evade interception.

We have also limited our experiments to those in which guard drones are faster than bandits; situations with faster bandits have been left out of the scope of this paper for several reasons. First, bandits do not perceive or react to guards; therefore, guards are incapable of any kind of “shepherding” of bandits, regardless of relative speeds. This means that all interceptions of faster-bandits must be done head-on or cross-range so as to achieve a positive closure speed; however, these geometries will lead to very high relative velocities (e.g. 100s of meters per second). For extremely high relative velocities, the current modeling of interception adjudication is unrealistic and any simulation results produced would just be artifacts of these unrealistic simulation assumptions; not reflections of real-world performance. Extensive, high-fidelity platform and sensor models would be needed to make them realistic. In other words, the only thing an overmatched guard team could do is throw many interceptors head-first at a fast bandit until one of them happens to achieve an “interception”, but that simulated interception would not be based on any realistic physics; and thus there is nothing to be learned from this exercise.

Acknowledgements

The authors thank David Fridovich-Keil and Kushagra Gupta at The University of Texas at Austin for helpful discussions and feedback on this work.

A Appendices

A.1 Discounted-Redundancy Assignment Reformulation

The discounted-redundancy objective in Sec. 2.2.1 can be represented as a rectangular linear assignment problem. For every active bandit j , create N_{G_A} virtual assignment slots (j, k) , with $k \in \{1, \dots, N_{G_A}\}$. The k th slot represents the k th guard assigned to bandit j and has marginal value $v_j \beta^{k-1}$. Let $y_{i,j,k} \in \{0, 1\}$ indicate whether guard i is assigned to slot (j, k) , and define the corresponding linear cost as

$$d_{i,j,k}^{\text{TA}} = c_{ij}^{\text{TA}} - v_j \beta^{k-1}. \quad (22)$$

The resulting assignment problem is

$$\min_{y_{i,j,k}} \sum_{i \in G_A} \sum_{j \in B_A} \sum_{k=1}^{N_{G_A}} d_{i,j,k}^{\text{TA}} y_{i,j,k}, \quad (23)$$

$$\text{subject to} \quad \sum_{j \in B_A} \sum_{k=1}^{N_{G_A}} y_{i,j,k} = 1, \quad \forall i \in G_A, \quad (24)$$

$$\sum_{i \in G_A} y_{i,j,k} \leq 1, \quad \forall j \in B_A, k \in \{1, \dots, N_{G_A}\}, \quad (25)$$

$$y_{i,j,k} \in \{0, 1\}. \quad (26)$$

There are N_{G_A} guard rows and $N_{B_A} N_{G_A}$ virtual bandit-slot columns. Although stated with binary variables, the linear-program relaxation is integral because the bipartite assignment constraint matrix is totally unimodular. Since slot values weakly decrease with k , an optimal solution can always be interpreted as filling earlier slots for a bandit before later slots. The cost matrix can be solved directly using SciPy's `scipy.optimize.linear_sum_assignment` routine.

References

- [1] Ross E. Allen. PYDGENS: Python/JAX differential game equilibria numerical solvers. <https://github.com/mit-ll/pydgens>, 2026. Accessed: 2026-09-01.
- [2] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*, volume 23 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2 edition, 1999.
- [3] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D. Dragan, and Claire J. Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481. IEEE, 2020.
- [4] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, Boca Raton, FL, 3 edition, 2013.
- [5] Simon Le Cleac’h, Mac Schwager, and Zachary Manchester. ALGAMES: A fast augmented lagrangian solver for constrained dynamic games. *Autonomous Robots*, 46(1):201–215, 2022.
- [6] Neil F. Palumbo, Ross A. Blauwkamp, and Justin M. Lloyd. Basic principles of homing guidance. *Johns Hopkins APL Technical Digest*, 29(1):25–41, 2010.
- [7] Ilan Rusnak. The lady, the bandits and the body guards—a two team dynamic game. *IFAC Proceedings Volumes*, 38(1):441–446, 2005.
- [8] J. K. Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S. Santos, et al. PettingZoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.