Aerial Swarm Defense using Interception and Herding Strategies

Vishnu S. Chipade and Dimitra Panagou

***Abstract*—This paper presents a multi-mode solution to the problem of defending a circular protected area (target) from a wide range of attacks by swarms of *risk-taking* and/or *risk-averse* attacking agents (attackers). The proposed multi-mode solution combines two defense strategies, namely: 1) an interception strategy for a team of defenders to intercept multiple *risk-taking* attackers while ensuring that the defenders do not collide with each other, 2) a herding strategy to herd a swarm of *risk-averse* attackers to a safe area. In particular, we develop mixed integer programs (MIPs) and geometry-inspired heuristics to distribute and assign and/or reassign the defenders to interception and herding tasks under different spatiotemporal behaviors by the attackers such as splitting into smaller swarms to evade defenders easily or high-speed maneuvers by some risk-taking attackers to maximize damage to the protected area. We provide theoretical as well as numerical comparison of the computational costs of these MIPs and the heuristics, and demonstrate the overall approach in simulations.**

***Index Terms*—autonomous agents, cooperative robots, task assignment, and multi-robot systems.**

I. Introduction

A. Motivation

Swarm technology has a wide range of applications [1], however may also pose threat to safety-critical infrastructure such as government facilities, airports, and military bases. The presence of adversarial agents or swarms nearby such entities, with the aim of causing physical damage or collecting critical information, can lead to catastrophic consequences. The adversarial agents (attackers) could be either risk-averse (self-interested), or risk-taking. Risk-averse attackers will try to avoid collision with other static or dynamic agents in order to avoid any damage to themselves. Risk-averse attackers could be more interested in collecting critical information by loitering around the safety-critical area (protected area) than intending to physically damage the protected area. On the other hand, risk-taking attackers will have low priority for their own survival compared to their mission. Such attackers could be interested in physically damaging the protected area. The degree of risk-aversion could vary among the attackers. Furthermore, the attackers may 1) cooperate among themselves and stay together as a swarm or do not stay together, or 2) do not cooperate among themselves.

Research has attributed various defense strategies to defend against different types of attackers, for example, 1) physical interception strategies [2]–[12] (mostly for risk-taking attackers), 2) herding strategies [13]–[23] (mostly against risk-averse attackers). With a wide range of potential behaviors by the attackers, a single type of defense approach may not be sufficient, economical or even desirable. In this paper, we combine interception-based and herding-based defense strategies for the defenders to provide a multi-mode defense solution against a wide range of adversarial attacks.

B. Related work

1) Multi-player pursuit evasion games: In pursuit-evasion games a team of pursuers aims to capture or intercept a team of evaders, while the evaders aim to evade from pursuers for as long as possible. Various approaches including optimal control techniques [24], area-minimization techniques [4], [25], value function based technique [26], mean-field approach and reinforcement-learning techniques [27], [28] exist in the literature to solve pursuit-evasion games. The existing solutions provide useful insights, however they in principle do not consider an area under risk that is targeted by the attackers. Therefore, pursuit-evasion approaches are less suitable for the class of area-defense problems studied in this paper.

2) Multi-agent area (target) defense: The area or target defense problem with a single agent on either team has been studied as a zero-sum differential game using various solution techniques including optimal control [29]–[33] and reachability analysis [34]. However, extending these approaches to multi-agent settings suffers from the curse of dimensionality. To remedy this, researches have been using a "divide and conquer" approach, i.e., solve the one-on-one problem or the problem with small number of agents for all such combinations of the agents, and scale up this solution to the original multi-agent problem.

In [3], the authors consider a multiplayer reach-avoid game. The authors solve the reach-avoid game for each pair of defender and attacker operating in a compact domain with obstacles using a Hamilton-Jacobi-Issacs (HJI) reachability approach. The solution is then used to assign defenders against the attackers using graph-theoretic maximum matching.

In the perimeter defense problem studied in [10] defenders are restricted to move on the perimeter of a protected

The first author is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA; the second author is with the Department of Robotics and the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA. (vishnuc, dpanagou)@umich.edu

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Award No. 1738714 along with significant contributions from C-UAS industry members.

area. Local games between small teams of defenders and attackers are solved and then assignments are done using a polynomial time algorithm.

The aforementioned studies provide useful insights to the area or target defense problem, however, are limited in application due to the use of simple motion models, such as single integrators. In [5], Target-Attacker-Defender (TAD) game with agents moving under double-integrator dynamics is considered. Due to the increased computational complexity of solving a zero-sum differential game optimally for high-dimensional systems, the authors use an isochrones method to design time-optimal control strategies for the players in 1-vs-1 TAD game. However, despite bounded acceleration inputs, no bounded velocities for the agents can be ensured or is assumed in [5].

In all of the aforementioned work, the defenders coordinate with each other for the assignment task to intercept the attackers, however, they do not consider collision avoidance among themshelves. Furthermore, the aforementioned interception strategies, while useful against *risk-taking* attackers, may be an extreme measure against *risk-averse* attackers. In other words, there may be cases where one may prefer to herd the *risk-averse* defenders to some safe area and take control of these attackers in favor of the defenders, instead of intercepting them.

3) Swarm herding: Herding has been studied previously in [13]–[15]. The approach in [13] uses an n-wavefront algorithm to herd a flock of birds away from an airport, where the birds on the boundary of the flock are influenced based on the locations of the airport and a safe area.

The herding method in [14] utilizes a circular-arc formation of herders to influence the nonlinear dynamics of the herd based on a potential-field approach, and designs a point-offset controller to guide the herd close to a specified location. In [15], biologically-inspired strategies are developed for confining a group of agents; the authors develop strategies based on the "wall" and "encirclement" methods that dolphins use to capture a school of fish. In addition, they compute regions from which this confinement is possible; however, the results are limited to constant-velocity motion. A similar approach called *herding by caging* is adopted in [16], where a cage of high potential is formed around the attackers. An RRT approach is used to find a motion plan for the agents; however, the cage is assumed to have already been formed around the agents, while the caging of the agents thereafter is only ensured with constant velocity motion under additional assumptions on the distances between the agents. Forming such a cage could be more challenging in case of self-interested, risk-averse attackers under non-constant velocity motion.

In [17], [18], the authors discuss herding using a switched-system approach; the herder (defender) chases targets (evaders/attackers) sequentially by switching among them so that certain dwell-time conditions are satisfied to guarantee stability of the resulting trajectories. However, the assumption that only one of the targets is influenced by the herder at any time might be limiting and non-practical in real applications. The authors in [19] use approximate dynamic programming to obtain suboptimal control policies for the herder to chase a target agent to a goal location. A game-theoretic formulation is used in [20] to address the herding problem by constructing a virtual barrier similar to [14]. However, the computational complexity due to the discretization of the state and control-action spaces limits its applicability.

Most of the aforementioned approaches for herding are limiting due to one or many of the following aspects: 1) simplified motion models, 2) absence of obstacles in the environment, 3) no consideration of inter-agent collisions, 4) assumption of a particular form of potential field to model the repulsive motion of the attackers with respect to the defenders.

We have addressed the above issues in our recent work [21], [22], which develops a method, termed as 'StringNet Herding', for defending a protected area from a swarm of attackers in a 2D obstacle environment. In 'StringNet Herding', a closed formation of strings ('StringNet') is formed by the defenders to surround the swarm of attackers. It is assumed that the attackers will stay together within a circular footprint as a swarm and collectively avoid the defenders. It is also assumed that the string between two defenders serves as a barrier through which the attackers cannot escape (e.g., a physical straight-line barrier, or some other mechanism). The StringNet is then controlled to herd the swarm of attackers to a safe area. The control strategy for the defenders in 'StringNet Herding' is a combination of time-optimal control actions and finite-time, state-feedback, bounded control actions, so that the attackers can be herded to safe area in a timely manner.

In [23], [35], we extended the 'StringNet Herding' approach to scenarios where attackers no longer stay together and may split into smaller swarms in reaction to the defenders' presence. Particularly, we first identify the spatial distributions (clusters/swarms) of the attackers that satisfy certain properties, using the density-based spatial clustering for applications with noises (DBSCAN) algorithm [36]. Then, we developed a mixed-integer quadratically constrained program (MIQCP) to distribute and assign the sub-teams of the defenders to the identified clusters of the attackers, so that the clusters of the attackers are herded to one of the safe areas. Note that we use swarm and cluster interchangeably throughout the paper.

C. Overview of the proposed approach

As discussed above, a wide range of approaches exist for area defense scenarios. However, only a specific type of behavior by the attackers is considered in each of the aforementioned works. To address a wide range of behaviors by the attackers a multi-mode solution is provided in this paper. We first make the following assumption.

Assumption 1 (Inter-Defender Collision-Aware Interception Strategy (IDCAIS))**.** *There exists an interception strategy to intercept multiple attackers in an area-defense game, such that the defenders account for inter-defender*

collisions while they intercept the attackers as quickly as possible.

Such interception strategy is provided in [12] (under review).

The multi-mode defense approach discussed in this paper is summarized in Figure 1. In this multi-mode defense approach, the spatial distributions of the attackers are continuously monitored using the DBSCAN algorithm, which classifies attackers into clusters of at least three agents. The attackers that either belong to clusters of less than 3 attackers, or are classified as noises by the DBSCAN algorithm, are called unclustered attackers. At time $t = 0$ s (the right half section in Figure 1), the defending team employs the IDCAIS against the unclustered attackers; under this interception strategy, some of the defenders are assigned to intercept the unclustered attackers in minimum time using collision-aware defender-to-attacker assignment (CADAA) [12] (discussed later), these defenders are called intercepting defenders. The rest unassigned defenders, called herding defenders, are distributed into sub-teams and assigned to herd the identified clusters of the attackers to one of the safe areas using 'StringNet Herding' approach [23], as long as the attackers stay together and avoid the defenders. If the attackers further split into new smaller clusters and/or individual attackers (unclustered attackers) at some time $t > 0$ (shown in the left half section in Figure 1), then the defenders are also further distributed into smaller sub-teams and assigned to herd the newly formed attackers' clusters and to intercept the newly-identified unclustered attackers that separated from the original cluster of the attackers using an optimal assignment algorithm.

D. Summary of our contributions

We develop a multi-mode defense strategy against wide range of swarm attacks using the IDCAIS and the 'StringNet Herding' [22] approach. Compared to the prior literature and our own work, the contributions of this paper are:

1) a centralized, iterative algorithm to assign the defenders to the attackers' clusters identified at $t = 0$ so that the defenders gather on the shortest paths of the attackers' swarms to the protected area before the attackers reach there;
2) a decentralized algorithm using mixed integer quadratically constrained quadratic programs (MIQCQPs) to assign the defenders to intercept the unclustered attackers, and to herd the attackers' newly-formed swarms in the case a swarm of attackers splits into smaller swarms at any future time $t > 0$;
3) heuristics to solve the MIQCQP approximately but quickly to find the assignment in real time;
4) theoretical as well as numerical comparison of the computational cost of the assignment algorithms.

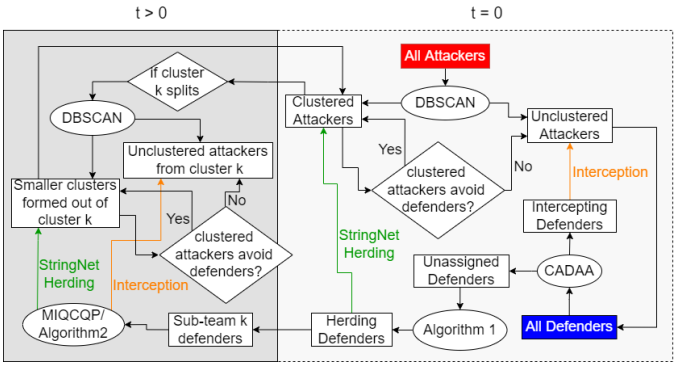


Figure 1: Overview of the Multi-mode Defense Approach

E. Organization

The rest of the paper is structured as follows. Section II provides the mathematical modeling, assumptions made and a statement of the problem studied. The strategy and the assignment algorithms of the multi-mode defense approach are discussed in Section III.More specifically, the optimal assignment algorithms at $t = 0$ and $t > 0$, their sub-optimal but computationally better alternative algorithms, heuristics to solve these optimal assignment problems in a computationally efficient manner, as well as their performance comparison are discussed in Section III. Simulation results for various scenarios demonstrating the proposed multi-mode framework are provided in Section IV. The paper is concluded in Section V.

II. Modeling and Problem Statement

Notation: We use $\|\cdot\|$ to denote the Euclidean norm of its argument. $|\cdot|$ denotes absolute value of a scalar argument or cardinality of a set argument. A ball of radius ρ centered at the origin is defined as $\mathcal{B}_\rho = \{\mathbf{r} \in \mathbb{R}^2 | \, \|\mathbf{r}\| \leq \rho\}$ and that centered at $\mathbf{r}_c$ is defined $\mathcal{B}_\rho(\mathbf{r}_c) = \{\mathbf{r} \in \mathbb{R}^2 | \, \|\mathbf{r} - \mathbf{r}_c\| \leq \rho\}$. $A \backslash B$ denotes all the elements of the set A that are not in the set B. Some most commonly used variables in the paper are described in Table I.

We consider N_a attackers denoted as $\mathcal{A}_i$, $i \in I_a = \{1, 2, ..., N_a\}$, and N_d defenders denoted as $\mathcal{D}_j$, $j \in I_d = \{1, 2, ..., N_d\}$, operating in a 2D environment $\mathcal{W} \subseteq \mathbb{R}^2$ that contains a protected area $\mathcal{P} \subset \mathcal{W}$, defined as $\mathcal{P} = \{\mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r}\| \leq \rho_p\}$, and N_s safe areas $\mathcal{S}_m \subset \mathcal{W}$, defined as $\mathcal{S}_m = \{\mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r} - \mathbf{r}_{sm}\| \leq \rho_{sm}\}$, for all $m \in I_s = \{1, 2, ..., N_s\}$, where ρ_p and ρ_{sm} are the radii of the protected area and m^{th} safe area, respectively, and $\mathbf{r}_{sm}$ is the center of m^{th} safe area. Visual depiction of the above elements is shown in Figure 2. The number of defenders is no less than that of attackers, i.e., $N_d \geq N_a$. The agents $\mathcal{A}_i$ and $\mathcal{D}_j$ are modeled as discs of radii ρ_a and ρ_d, where $\rho_d \leq \rho_a$, respectively. Let $\mathbf{r}_{ai} = [x_{ai} \; y_{ai}]^T$ and $\mathbf{r}_{dj} = [x_{dj} \; y_{dj}]^T$ be the position vectors of $\mathcal{A}_i$ and $\mathcal{D}_j$, respectively; $\mathbf{v}_{ai} = [v_{x_{ai}} \; v_{y_{ai}}]^T$, $\mathbf{v}_{dj} = [v_{x_{dj}} \; v_{y_{dj}}]^T$ be the velocity vectors, respectively, and $\mathbf{u}_{ai} = [u_{x_{ai}} \; u_{y_{ai}}]^T$, $\mathbf{u}_{dj} = [u_{x_{dj}} \; u_{y_{dj}}]^T$ be the accelerations, which serve also as the control inputs, respectively, all resolved in a global inertial frame $\mathcal{F}_{gi}(\hat{\mathbf{i}}, \hat{\mathbf{j}})$ (see Fig.2). The agents move under

Table I: Table of notation

$\mathcal{A}_i$	denotes the i^{th} attacker
$\mathcal{A}_{c_k}(t)$	denotes the group of attackers indexed by $A_{c_k}(t)$
$A_{c_k}(t)$	set of indices of the attackers in k^{th} cluster of attackers at time t
$A_{uc}(t)$	set of indices of the unclustered attackers at time t
$A_c^{(k)}(t)$	set of indices of clusters of the attackers separated from the k^{th} cluster of attackers at time t
$A_{uc}^{(k)}(t)$	set of indices of the unclustered attackers separated from the k^{th} cluster of attackers at time t
$\mathscr{A}_k(t_{se})$	data structure storing information of the attackers in k^{th} cluster of attackers after it splits at $t = t_{se}$
$\mathscr{A}_k(t_{se}).f$	denotes the data field f of the data structure $\mathscr{A}_k(t_{se})$ at time $t = t_{se}$
$\mathcal{D}_j$	denotes the j^{th} defender
$\mathcal{D}_{c_k}(t)$	denotes the group of defenders indexed by $D_{c_k}(t)$
$\mathcal{D}_{c_k}^c(t)$, $\mathcal{D}_{c_k}^t(t)$	group of central and terminal defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t))$, resp.
$\mathcal{D}_{c_k}^l(t)$, $\mathcal{D}_{c_k}^r(t)$	group of terminal defenders on the left and right end of Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t))$, resp.
$D_{c_k}(t)$	set of indices of the defenders assigned to k^{th} cluster of attackers at time t
$\mathscr{D}_k(t_{se})$	data structure storing information of the defenders indexed by $D_k(t_{se}^-)$
$\mathscr{D}_k(t_{se}).f$	denotes the data field f of the data structure $\mathscr{D}_k(t_{se})$ at time $t = t_{se}$
$\mathcal{G}_{sn}^{cl}(I_d)$	Closed-StringNet formed by the defenders with indices as in the set I_d
$\mathcal{G}_{sn}^{op}(I_d)$	Open-StringNet formed by the defenders with indices as in the set I_d
I_a, $I_{ac}(t)$	equals set $\{1, 2, ..., N_a\}$, $\{1, 2, ..., N_{ac}(t)\|)\}$, resp.
I_d, $I_{dc_k}(t)$	equals set $\{1, 2, ..., N_d\}$, $\{1, 2, ..., \mathscr{R}_d(\|A_{c_k}(t)\|)\}$, resp.
N_a, $N_{ac}(t)$	number of attackers and attackers' clusters, resp.
N_d,	number of defenders
$\mathscr{R}_d(\cdot)$	defender-to-attacker resource allocation function
$\mathbf{r}_{ai}$, $\mathbf{r}_{dj}$	position of i^{th} attacker, j^{th} defender, resp.
$\mathbf{r}_{sm}$	center m^{th} safe area
t_{se}	time at which attackers' split event happens
t_{se}^-	time instant just before attackers' split event
$\mathbf{u}_{ai}$, $\mathbf{u}_{dj}$	acceleration of i^{th} attacker, j^{th} defender, resp.
$\mathbf{v}_{ai}$, $\mathbf{v}_{dj}$	velocity of i^{th} attacker, j^{th} defender, resp.
$\beta_c(t)$	set of mappings of defenders' assignment to the clusters of the attackers at time t
$\beta_{c_k}(t)$	mapping that assigns defenders to the k^{th} cluster of the attackers at time t
$\beta_{uc}(t)$	mapping that assigns defenders to the unclustered attackers at time t
ρ_d^{int}	interception radius of a defender
$\delta_{jk}^{herd}(t)$	decision variable to decide if $\mathcal{D}_j$ is assigned to herd attackers' swarm $\mathcal{A}_{c_k}$ at time t
$\delta_{ji}^{int}(t)$	decision variable to decide if $\mathcal{D}_j$ is assigned to intercept the attacker $\mathcal{A}_i$ at time t

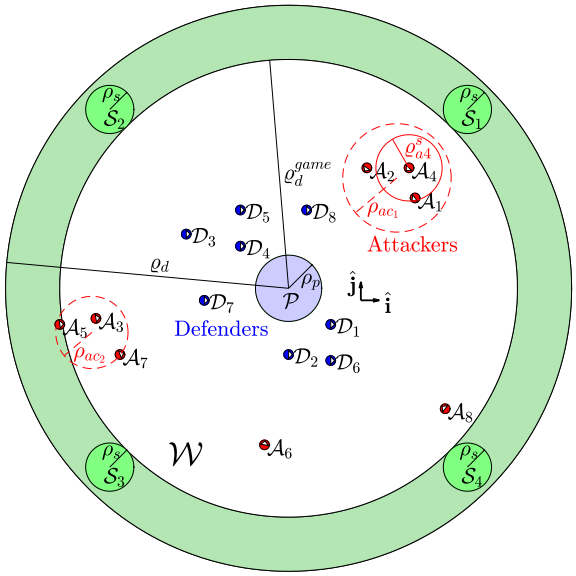


Figure 2: Schematic of a scenario showing multiple attackers (red filled circles with white arrows), some as risk-averse swarms while some individual risk-taking attackers, trying to reach the protected area $\mathcal{P}$ and defenders (blue filled circles with white arrows) spread around $\mathcal{P}$.

double integrator (DI) dynamics with linear drag (damped double integrator), similar to isotropic rocket [37]:

$$\dot{\mathbf{x}}_\star = \begin{bmatrix} \dot{\mathbf{r}}_\star \\ \dot{\mathbf{v}}_\star \end{bmatrix} = \begin{bmatrix} \mathbf{0}_2 & \mathbf{I}_2 \\ \mathbf{0}_2 & -C_D\mathbf{I}_2 \end{bmatrix} \mathbf{x}_\star + \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{I}_2 \end{bmatrix} \mathbf{u}_\star \tag{1}$$

where $\star \in \{ai | i \in I_a\} \cup \{dj | j \in I_d\}$, $C_D > 0$ is the known, constant drag coefficient. The accelerations $\mathbf{u}_{ai}$ and $\mathbf{u}_{dj}$ are bounded by $\bar{u}_a$, $\bar{u}_d$ as given in (2) such that $\bar{u}_a < \bar{u}_d$.

$$\|\mathbf{u}_{ai}\| \leq \bar{u}_a, \quad \|\mathbf{u}_{dj}\| \leq \bar{u}_d, \tag{2}$$

By incorporating the drag term, the damped double integrator (1) inherently poses a speed bound on each agent under a limited acceleration control, i.e., $\|\mathbf{v}_{ai}\| < \bar{v}_a = \frac{\bar{u}_a}{C_D}$ and $\|\mathbf{v}_{dj}\| < \bar{v}_d = \frac{\bar{u}_d}{C_D}$, and does not require an explicit constraint on the velocity of the agents while designing bounded controllers, as in earlier literature. So we have $\mathbf{x}_{ai} \in \mathcal{X}_a$, for all $i \in I_a$, where $\mathcal{X}_a = \mathbb{R}^2 \times \mathcal{B}_{\bar{v}_a}$ and $\mathbf{x}_{dj} \in \mathcal{X}_d$, for all $j \in I_d$, where $\mathcal{X}_d = \mathbb{R}^2 \times \mathcal{B}_{\bar{v}_d}$. We make the following assumption:

Assumption 2. *All the defenders know the position $\mathbf{r}_{ai}$ and velocity $\mathbf{v}_{ai}$ of the attacker $\mathcal{A}_i$ that lies inside a circular sensing zone $\mathcal{Z}_d = \{\mathbf{r} \in \mathbb{R}^2 | \ \|\mathbf{r}\| \leq \varrho_d\}$ for all $i \in I_a$, where $\varrho_d > 0$ is the radius of the defenders' sensing zone. Every attacker $\mathcal{A}_i$ has a similar local sensing zone $\mathcal{Z}_{ai} = \{\mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r} - \mathbf{r}_{ai}\| \leq \varrho_{ai}\}$, where $\varrho_{ai} > 0$ is the radius of $\mathcal{A}_i$'s sensing zone (Fig. 2).*

For Assumption 2 to hold, a system of sensors such as radars, lidars, cameras, etc., that are spatially distributed around the protected area can be used. The data from these sensors are assumed to be processed by a central computer and communicated to all the defenders.

Each defender is capable of connecting to other two defenders via string barriers. String barriers are realized as impenetrable and extendable line barriers (e.g., spring-loaded pulley and a rope or other similar mechanism [38]) that prevent attackers from passing through them. The extendable string barrier allows free relative motion of the two defenders connected by the string. The string barrier can have a maximum length of $\bar{R}_{sb}$. If the string barrier were to be physical one, then it can be established between two defenders $\mathcal{D}_j$ and $\mathcal{D}_{j'}$ only when they are close to each other and have almost same velocity, i.e., $\|\mathbf{r}_{dj} - \mathbf{r}_{dj'}\| \leq \epsilon_1 < \bar{R}_{sb}$ and $\|\mathbf{v}_{dj} - \mathbf{v}_{dj'}\| \leq \epsilon_2$, where ϵ_1 and ϵ_2 are small numbers that depend on the physical size of the defenders as well as the mechanism and their capability to physically connect at a given distance.. Each defender $\mathcal{D}_j$

is endowed with an interception/capture radius ρ_d^{int}, i.e., the defender $\mathcal{D}_j$ is able to physically damage an attacker $\mathcal{A}_i$ when $\|\mathbf{r}_{dj}(t) - \mathbf{r}_{ai}(t)\| < \rho_d^{int}$ for some $t > 0$.

The goal of the attackers is to send as many attackers as possible to the protected area $\mathcal{P}$. The defenders aim to either intercept these attackers or herd them away to one of the safe areas in $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{N_s}\}$ in order to defend the protected area $\mathcal{P}$. Formally, we consider the following problem.

Problem 1 (Swarm Defense)**.** *Design a defense strategy for a team of defenders to defend a protected area from a wide range of adversarial attacks by attackers, where attackers could possibly stay together as swarms or stay alone during the attack.*

Next, we discuss the multi-mode defense strategy that addresses Problem 1.

III. Multi-mode Defense Strategy

The attackers may show wide range of behaviors, such as: some or all attackers staying close together, some or all attackers avoiding defenders while attacking the protected area, some attackers not intending to damage the protected area but only interested in reaching in its neighborhood maybe for collecting some key information, while some attackers only interested in physically damaging the protected area at any cost, etc.

In this section, we provide a multi-mode algorithm to combine the 'StringNet Herding' approach developed in [21]–[23] and the IDCAIS to defend against wide range of behaviors by the attackers discussed earlier. In the following, we first revisit some key definitions related to 'StringNet Herding'.

Definition 1 (Closed-StringNet)**.** *The Closed-StringNet $\mathcal{G}_{sn}^{cl}(I_d) = (\mathcal{V}_{sn}^{cl}(I_d), \mathcal{E}_{sn}^{cl}(I_d))$ is a cycle graph consisting of: 1) a subset of defenders as the vertices, $\mathcal{V}_{sn}^{cl}(I_d) = \{\mathcal{D}_j \mid j \in I_d\}$, 2) a set of edges, $\mathcal{E}_{sn}^{cl}(I_d) = \{(\mathcal{D}_j, \mathcal{D}_{j'}) \in \mathcal{V}_{sn}^{cl}(I_d) \times \mathcal{V}_{sn}^{cl}(I_d) | \mathcal{D}_j \overset{s}{\longleftrightarrow} \mathcal{D}_{j'}\}$, where the operator $\overset{s}{\longleftrightarrow}$ denotes an impenetrable line barrier between the defenders.*

Definition 2 (Open-StringNet)**.** *The Open-StringNet $\mathcal{G}_{sn}^{op}(I_d) = (\mathcal{V}_{sn}^{op}(I_d), \mathcal{E}_{sn}^{op}(I_d))$ is a path graph consisting of: 1) a set of vertices, $\mathcal{V}_{sn}^{op}(I_d)$ and 2) a set of edges, $\mathcal{E}_{sn}^{op}(I_d)$, similar to that in Definition 1.*

The 'StringNet Herding' approach consists of four phases: 1) gathering, 2) seeking, 3) enclosing, and 4) herding. In the gathering phase, the defenders establish an Open-StringNet on the time-optimal path of the attackers' swarm. Then in the seeking phase they seek to get close to the attackers' swarm if they had not traveled along their time-optimal trajectory as expected by the defenders. During the seeking phase, the defenders ensure that they maintain the Open-StringNet formation. Next, during the enclosing phase, as the defenders come sufficiently close to the attackers, they enclose the attackers by establishing a Closed-StringNet around the attackers' swarm. In the herding phase, the Closed-StringNet is moved to the nearest safe area which also takes the enclosed attackers to the safe area.

Next, we describe how the defenders are assigned to either intercept the unclustered (more likely risk-taking) attackers or herd the clustered (more likely risk-averse) attackers during different temporal and spatial events.

A. Optimal assignment at $t = 0$

We first identify spatial distributions (clusters) of the attackers that are detected in the annular region between the circles $\mathbf{r} = \varrho_d$ and $\mathbf{r} = \varrho_d^{game}$ (see Fig. 2). For the cluster identification, we use DBSCAN algorithm [36] with parameters $\varepsilon_{nb} = \frac{\bar{\rho}_{ac}(m_{pts}-1)}{N_a-1}$ and $m_{pts} = 3$ where $\bar{\rho}_{ac} = \frac{\bar{R}_{sb}}{2}\cot(\frac{\pi}{N_d})$ is the radius of the largest circle inscribed in the largest Closed-StringNet formation that can be formed by the N_d defenders. This choice of parameters for the DBSCAN algorithm ensures that the identified clusters have more than 3 attackers in them and have sizes for which subteams of the defenders can be found which can herd these clusters. This is because one needs at least 3 defenders to form a Closed-StringNet and if $N_d = N_a$ then we may not have enough defenders to enclose all swarms of the attackers with less than 3 attackers in them. Hence all the swarms of the attackers with less than 3 attackers will be termed as singular swarms and the member attackers of these singular swarms will be identified as noise by DBSCAN algorithm and classified as unclustered attackers. For more details on how the parameters of the DBSCAN are chosen, refer to [23]. Let $\mathcal{A}_c(0) = \{\mathcal{A}_{c_1}(0), \mathcal{A}_{c_2}(0), \ldots, \mathcal{A}_{c_{N_{ac}(0)}}(0)\}$ be the set of $N_{ac}(0)$ swarms of the attackers at $t = 0$ identified using the DBSCAN algorithm. Here $\mathcal{A}_{c_k}(0) = \{\mathcal{A}_i | i \in A_{c_k}(0)\}$, for $k \in I_{ac}(0) = \{1, 2, 3, ..., N_{ac}(0)\}$ where $A_{c_k}(0) \subseteq I_a$ is the set of indices of the attackers that belong to the k^{th} cluster of the attackers at $t = 0$. Let $\mathcal{A}_{uc}(0) = \{\mathcal{A}_i | i \in A_{uc}(0)\}$ denote the set of unclustered attackers where $A_{uc}(0) \subseteq I_a$ is the set of indices of the attackers that are not clustered by the DBSCAN algorithm, i.e., the attackers that are treated as the noises by the DBSCAN algorithm. The defenders aim to intercept the unclustered attackers assuming that these attackers are risk-taking while they attempt to herd the clustered attackers with the hope that the clustered attackers will stay together and try to avoid the defenders. For this we need to assign some individual defenders to intercept the unclustered attackers and some sub-teams of the defenders to herd the identified clusters of the attackers. Since the unclustered attackers are likely to be risk-taking and hence pose more risk to the protected area, the assignment of the best defenders to intercept these unclustered attackers is done first and then the rest defenders are assigned to herd the clustered attackers.

We first us collision-aware defender-to-attacker assignment (CADAA) to assign defenders to intercept the identified unclustered attackers $\mathcal{A}_{uc}(0)$ such that these attackers are intercepted as quickly as possible and the possible collisions among the defenders are minimized. Let $\delta_{ji}^{int}(0)$ be the binary decision variable at time $t = 0$ that takes

value 1 if the defender $\mathcal{D}_j$ is assigned to intercept attacker $\mathcal{A}_i$ and 0 otherwise. Let $C_d^{int}(\mathbf{X}_{dj}^{ai})$ be the cost incurred by the defender $\mathcal{D}_j$ to capture the attacker $\mathcal{A}_i$ and is given by:

$$C_d^{int}(\mathbf{X}_{dj}^{ai}) = \begin{cases} t_d^{int}(\mathbf{x}_{dj}, \mathbf{x}_{ai}), & \text{if } \mathbf{x}_{ai} \in \mathcal{R}_d(\mathbf{x}_{dj}); \\ c_l, & \text{otherwise;} \end{cases} \tag{3}$$

where $\mathbf{X}_{dj}^{ai} = [\mathbf{x}_{dj}^T, \mathbf{x}_{ai}^T]^T$, $t_d^{int}(\mathbf{x}_{dj}, \mathbf{x}_{ai})$ is the minimum time required by the defender $\mathcal{D}_j$ to capture the attacker $\mathcal{A}_i$ that is moving towards the protected area $\mathcal{P}$ under time-optimal control action as defined in [12], c_l ($>> 1$) is a very large number, and $\mathcal{R}_d(\mathbf{x}_{dj}) = \{\mathbf{x}_a \in \bar{\mathcal{X}}_a | t_d^{int}(\mathbf{x}_d, \mathbf{x}_a) - t_a^{int}(\mathbf{x}_a, \mathbf{r}_p) \leq 0\}$ is the winning region of the defender $\mathcal{D}_j$ starting at $\mathbf{x}_{dj}$, where $\bar{\mathcal{X}}_a = (\mathbb{R}^2 \backslash \mathcal{P}) \times \mathcal{B}_{\bar{v}_a}$, $t_a^{int}(\mathbf{x}_a, \mathbf{r}_p)$ is the time that the attacker starting at $\mathbf{x}_a$ requires to reach the protected area at $\mathbf{r}_p$. Let $C_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'})$ is the cost associated with a collision that may occur between the two defenders that are assigned interception task and is defined as:

$$C_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'}) = \begin{cases} \frac{1}{t_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'})}, & \text{if } \mathcal{D}_j \ \& \ \mathcal{D}_{j'} \text{ collide;} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $t_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'})$ is time of collision between $\mathcal{D}_j$ and $\mathcal{D}_{j'}$ on their time-optimal trajectories.

We find the optimal $\delta_{ji}^{int*}(0)$ by solving the following CADAA problem at $t = 0$:

$$\begin{aligned} \underset{\boldsymbol{\delta}^{int}(0)}{\arg\min} \sum_{i \in A_{uc}(0)} \sum_{j \in I_d} \Big(& (1-w) C_d^{int}(\mathbf{X}_{dj}^{ai}) \delta_{ji}^{int}(0) + \\ & w \sum_{i' \in A_{uc}(0)} \sum_{j' \in I_d} C_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'}) \delta_{ji}^{int}(0) \delta_{j'i'}^{int}(0) \Big) \end{aligned} \tag{5a}$$

$$\text{Subject to } \sum_{i \in A_{uc}(0)} \delta_{ji}^{int}(0) = 1, \quad \forall j \in I_d; \tag{5b}$$

$$\sum_{j \in I_d} \delta_{ji}(0) = 1, \quad \forall i \in A_{uc}(0); \tag{5c}$$

$$\delta_{ji}^{int}(0) \in \{0, 1\}, \quad \forall j \in I_d, \ i \in A_{uc}(0); \tag{5d}$$

where $\boldsymbol{\delta}^{int}(0) = [\delta_{ji}^{int}(0) | i \in A_{uc}(0), j \in I_d]^T \in \{0,1\}^{N_d |A_{uc}(0)|}$ is the binary decision vector and $w \in (0, 1)$ is user specified weight of the collision cost that is used to adjust the importance of the collisions among the defenders and the time to intercept the attackers at the assignment stage.

A mapping $\beta_{uc}(0, \cdot) : \{i \in A_{uc}(0)\} \to \{j \in I_d\}$, which gives the index of the defender assigned to intercept a given unclustered attacker $\mathcal{A}_i$ is then defined as:

$$\beta_{uc}(t, i) = \arg\max_j \delta_{ji}^{int*}(0), \quad \forall t \geq 0. \tag{6}$$

Let $\mathcal{D}_{uc}(0) = \{\mathcal{D}_{\beta_{uc}(t,i)} | i \in A_{uc}(0)\}$ denote the set of defenders that are assigned to the unclustered attackers $\mathcal{A}_{uc}(0)$ and $D_{uc}(0) = \{\beta_{uc}(t, i) | i \in A_{uc}(0)\}$ be the set of indices of the defenders in $\mathcal{D}_{uc}(0)$. Let $\mathcal{D}_c(0) = \{\mathcal{D}_j | j \in D_c(0)\}$ denote the set of all the other unassigned defenders, where $D_c(0) = I_d \backslash D_{uc}(0)$. These unassigned defenders $\mathcal{D}_c(0)$ are then employed to herd the identified clusters of the attackers.

Next, we describe a centralized approach to find a time-opimal, collision free motion plan for the defenders in $\mathcal{D}_c(0)$ to gather on the shortest paths of the attackers' swarms.

1) Centralized Approach: In this approach, the two problems: i) of choosing the best gathering formations, and ii) of the assignment of the defenders in $\mathcal{D}_c(0)$ to the goal locations on these gathering formations are solved simultaneously. We provide a bisection method based iterative scheme as detailed in Algorithm 1 to solve the above two problems simultaneously. Let $\mathscr{R}_d(N_a) : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ be the defender-to-attacker resource allocation function that outputs the number of the defenders that can be assigned to the given N_a attackers. We make the following assumption about the defender-to-attacker resource allocation function.

Assumption 3. *The defender-to-attacker resource allocation function is a strictly monotonically increasing function, i.e., $\mathscr{R}_d(N_a) < \mathscr{R}_d(N_a + 1)$, such that $\mathscr{R}_d(N_a) \geq N_a$.*

Assumption 3 ensures that there are adequate number of defenders to go after each attacker in the event the attackers in the swarm disintegrate into singular swarms.

Consider a line formation $\mathscr{F}_{dc_k}^{line}$ characterized by positions $\mathbf{p}_k^{line}(\mathbf{r}_{df_k}, \phi_k) = \{\mathbf{p}_{k,1}^{line}, \mathbf{p}_{k,2}^{line}, ..., \mathbf{p}_{k,\mathscr{R}_d(|A_{c_k}|)}^{line}\}$ where

$$\mathbf{p}_{k,l}^{line}(\mathbf{r}_{df_k}, \phi_k) \triangleq \mathbf{r}_{df_k} + \hat{R}_l \hat{\mathbf{o}}(\phi_k + \tfrac{\pi}{2}), \tag{7}$$

for all $l \in I_{dc_k}(0) = \{1, 2, ..., \mathscr{R}_d(|A_{c_k}(0)|)\}$, where $\hat{\mathbf{o}}(\theta) = [\cos(\theta), \sin(\theta)]^T$ is the unit vector making an angle θ with x-axis, and $\hat{R}_l = \hat{R}_d^{d,g}\left(\frac{\mathscr{R}_d(|A_{c_k}|) - 2l + 1}{2}\right)$, where $\hat{R}_d^{d,g} (\leq \bar{R}_{sb})$ is the user defined separation between the defenders at the gathering formation.

Corresponding to each attackers' cluster $\mathcal{A}_{c_k}$, the desired gathering formation $\mathscr{F}_{dc_k}^{g}$ for the defenders to gather at is chosen to be a line formation[1] $\mathscr{F}_{dc_k}^{line}$ centered at $\mathbf{r}_{df_k}$ with orientation ϕ_k, characterized by the positions $\boldsymbol{\xi}_{c_k}^g = \{\boldsymbol{\xi}_{c_k,1}^g, \boldsymbol{\xi}_{c_k,2}^g, ..., \boldsymbol{\xi}_{c_k,\mathscr{R}_d(|A_{c_k}|)}^g\} = \mathbf{p}_k^{line}(\mathbf{r}_{df_k}, \phi_k)$, as obtained in Algorithm 1. These positions are static, i.e., $\dot{\boldsymbol{\xi}}_{c_k,l}^g = \ddot{\boldsymbol{\xi}}_{c_k,l}^g = \mathbf{0}$ for all $l \in I_{dc_k}$. The gathering centers $\mathbf{r}_{df_k}$, for all $k \in I_{ac}(0)$, are chosen to lie outside the protected area $\mathcal{P}$. Algorithm 1 also outputs the Defender-to-Attacker-Swarm Assignment (DASA), β, which is defined formally as:

Definition 3 (Defender-to-Attacker-Swarm Assignment)**.** *A set $\beta_c(t) = \{\beta_{c_1}(t, \cdot), \beta_{c_2}(t, \cdot), ...\beta_{c_{N_{ac}(t)}}(t, \cdot)\}$ of mappings $\beta_{c_k}(t, \cdot) : \{1, 2, ..., \mathscr{R}_d(|\mathcal{A}_{c_k}(t)|)\} \to I_d$, where, for all $k \in I_{ac}(t)$, $\beta_{c_k}(t, l)$ gives the index of the defender, at time t, that is assigned to either gather at position $\boldsymbol{\xi}_{c_k,l}^g$ on the time-optimal path of swarm $\mathcal{A}_{c_k}(t)$ during the gathering phase, or track the desired position $\boldsymbol{\xi}_{c_k,l}^s$ or $\boldsymbol{\xi}_{c_k,l}^e$ or $\boldsymbol{\xi}_{c_k,l}^h$ during the seeking or enclosing or herding phase,*

[1]This is a better choice compared to a semicircular formation as chosen in [22]. Because, the semicircular formation, for a given length constraint on the string barrier ($\bar{R}_{sb}$), creates smaller blockage to the attackers as compared to the line formation. Although, Completing a circular formation starting from a semicircular formation of the same radius is faster. It is a trade-off between effectiveness and speed.

respectively, in order to successfully herd the swarm $\mathcal{A}_{c_k}(t)$ to the closest safe area.

The set of defenders assigned to gather on the path of the cluster $\mathcal{A}_{c_k}(0)$ is denoted by $\mathcal{D}_{c_k}(0) = \{\mathcal{D}_j | j \in D_{c_k}(0)\}$, where $D_{c_k}(0)$ is the set of indices defined as: $D_{c_k}(0) = \{\beta_{c_k}(0,1), \beta_{c_k}(0,2), ..., \beta_{c_k}(0, \mathscr{R}_d(0, |\mathcal{A}_{c_k}|))\}$ for all $k \in I_{ac}(0)$. Each of these sub-teams $\mathcal{D}_{c_k}(0)$'s of the defenders are tasked to achieve the Open-StringNet formations $\mathcal{G}^{op}_{sn}(D_{c_k}(0))$ on the shortest paths of the oncoming attacking swarms. Assuming $N_d = N_a$, we choose $\mathscr{R}_d(|\mathcal{A}_{c_k}|) = |\mathcal{A}_{c_k}|$, i.e., the number of defenders assigned to a swarm $\mathcal{A}_{c_k}$ is equal to the number of attackers in $\mathcal{A}_{c_k}$.

Algorithm 1: Gathering formations for the defenders

Input: $\mathbf{r}_d(0)$, $\mathbf{x}_a(0)$, $D_c(0)$, $\{A_{c_k}(0) | k \in I_{ac}(0)\}$

for $k = 1 : N_{ac}(0)$ **do**
  CoM of $\mathcal{A}_{c_k}(0)$: $\mathbf{x}_{ac_k}(0) = \sum_{i \in A_{c_k}(0)} \frac{\mathbf{x}_{ai}(0)}{|\mathcal{A}_{c_k}(0)|}$;
  $\mathbf{P}_{ac_k}$=`timeOptimalTraj` $(\mathbf{x}_{ac_k}(0))$;
while $\Sigma_{T_{lead}} > \epsilon_{tol}$ **do**
  $\Sigma_{T_{lead}} = 0; \gamma^{<}_{ac_k} = 0; \gamma^{>}_{ac_k} = \Gamma_{ac_k} - \rho_{pa}; \boldsymbol{\xi}^g = [\,]$;
  for $k = 1 : N_{ac}(0)$ **do**
    $\gamma_{ac_k} = \frac{\gamma^{<}_{ac_k} + \gamma^{>}_{ac_k}}{2}$;
    $\mathbf{r}_{df_k}(0) = \mathscr{P}_{ac_k}(\gamma_{ac_k})$;
    $\boldsymbol{\xi}^g_{c_k} = \mathbf{p}^{line}_k(\mathbf{r}_{df_k}(0), \vartheta_{ac_k}(\gamma_{ac_k}) - \pi)$
    $\boldsymbol{\xi}^g \leftarrow \{\boldsymbol{\xi}^g, \boldsymbol{\xi}^g_{c_k}\}$;
  $[\beta_c(0), \mathcal{T}]$=`assignDtoGMILP` $(\mathbf{r}_{dc}(0), \boldsymbol{\xi}^g)$;
  for $k = 1 : N_{ac}$ **do**
    $\Sigma_{T_{lead}} = \Sigma_{T_{lead}} + |\frac{\gamma_{ac_k}}{\bar{v}_a} - \mathcal{T}_k - \Delta T^g_{dc_k}|$;
    if $\frac{\gamma_{ac_k}}{\bar{v}_a} - \mathcal{T}_k - \Delta T^g_d < 0$ **then**
      $\gamma^{>}_{ac_k} = \gamma_{ac_k}$;
    else
      $\gamma^{<}_{ac_k} = \gamma_{ac_k}$;
return $\boldsymbol{\xi}^g$, $\beta_c(0)$, $\{\mathbf{r}_{df_1}(0), \mathbf{r}_{df_2}(0), ...\mathbf{r}_{df_{N_{ac}(0)}}(0)\}$

In Algorithm 1, `timeOptimalTraj`$(\mathbf{x}_{ac_k}(0))$ function finds the time-optimal trajectory $\mathbf{P}_{ac_k}$ for an agent starting at $\mathbf{x}_{ac_k}(0)$ to reach the protected area. The trajectory $\mathbf{P}_{ac_k}$ is associated with mappings $\mathscr{P}_{ac_k} : [0, \Gamma_{ac_k}] \to \mathbb{R}^2$ and $\vartheta_{ac_k} : [0, \Gamma_{ac_k}] \to [0, 2\pi]$. Here $\mathscr{P}_{ac_k}(\gamma_{ac_k})$ gives the Cartesian coordinates, and $\vartheta_{ac_k}(\gamma_{ac_k})$ gives the direction of the tangent to the path at the location reached after traveling γ_{ac_k} distance along the path from the initial position. $\mathbf{r}_d(0) = \{\mathbf{r}_{dj}(0) | j \in I_d\}$ is the set of initial positions of the defenders and $\mathbf{x}_a(0) = \{\mathbf{x}_{ai}(0) | i \in I_a\}$ is the set of initial states of the attackers. Each defender is assumed to have zero initial velocity[2]. The function `assignDtoGMILP` assigns each defender $\mathcal{D}_j$ in $\mathcal{D}_c(0)$ initially located at $\mathbf{r}_{dj}(0)$ to one of the gathering locations in $\boldsymbol{\xi}^g = \{\boldsymbol{\xi}^g_{c_1}, \boldsymbol{\xi}^g_{c_2}, ..., \boldsymbol{\xi}^g_{c_{N_{ac}(0)}}\}$ by solving the following the mixed integer linear program (MILP):

$$\arg\min_{\boldsymbol{\delta}} \sum_{k=1}^{N_{ac}(0)} \sum_{l=1}^{|I_{dc_k}(0)|} \sum_{j \in D_c(0)} \left\| \mathbf{r}_{dj}(0) - \boldsymbol{\xi}^g_{c_k,l} \right\| \delta^{c_k}_{jl} \quad (8a)$$

$$\text{Subject to } \sum_{k \in I_{ac}(0)} \sum_{l \in I_{dc_k}(0)} \delta^{c_k}_{jl} = 1, \quad \forall j \in D_c(0); \quad (8b)$$

$$\sum_{j \in D_c(0)} \delta^{c_k}_{jl} = 1, \quad \forall l \in I_{dc_k}(0), \forall k \in I_{ac}(0),; \quad (8c)$$

$$\delta^{c_k}_{jl} \in \{0,1\}, \quad \forall j \in D_c(0), \forall l \in I_{dc_k}(0), \forall k \in I_{ac}(0); \quad (8d)$$

where the distance between an initial position $\mathbf{r}_{dj}(0)$ and $\boldsymbol{\xi}^g_{c_k,l}$ is used as the metric for solving the assignment problem, the constraints (8b) ensure that each defender is assigned to a single goal location, the constraints (8c) ensure that each goal location is assigned a unique defender, and the last constraints (8d) force the decision variable $\delta^{c_k}_{jl}$ to be binary. The decision variable $\delta^{c_k}_{jl}$ is 1 if the defender $\mathcal{D}_j$ is assigned to go to the goal location $\boldsymbol{\xi}^g_{c_k,l}$ and 0 otherwise; and $\boldsymbol{\delta} \in \{0,1\}^{N_\delta(0)}$ is the binary decision vector defined as $\boldsymbol{\delta} = [\delta^{c_k}_{jl} | \forall j \in D_c(0),\ \forall l \in I_{dc_k}(0),\ \forall k \in I_{ac}(0)]^T$, where $N_\delta(0) = (N_d - |A_{uc}(0)|) \sum_{k \in I_{ac}(0)} \mathscr{R}_d(|A_{c_k}(0)|)$. The function `assignDtoGMILP` also outputs $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{N_{ac}}\}$, where $\mathcal{T}_k$, for all $k \in I_{ac}(0)$, is the time required by the sub-team $\mathcal{D}_{c_k}(0)$ to gather at their desired gathering formation. The parameter $\epsilon_{tol} > 0$ is a user defined small number used as the convergence tolerance.

The idea in Algorithm 1 is to find the gathering formations that are as far from the protected area as possible and each subteam $\mathcal{D}_{c_k}(0)$ of the defenders is able to reach their assigned gathering formation at least $\Delta T^g_{dc_k}$ s before the center of mass (CoM) of $\mathcal{A}_{c_k}$, that follows its time-optimal trajectory towards the protected area, reaches the center of the gathering formation. Here $\Delta T^g_{dc_k}$, for all $k \in I_{ac}(0)$ is a user-defined time that accounts for the size of the swarm $\mathcal{A}_{c_k}$ and the time required to get connected by strings once arrived at the desired formation.

The Defender-to-Attacker-Swarm Assignment $\beta_{c_k}(0, \cdot)$, for all $k \in I_{ac}(0)$, is then obtained as:

$$\beta_{c_k}(0, l) = \arg\max_j \delta^{c_k *}_{jl} \quad (9)$$

where $\delta^{c_k *}_{jl}$ is the optimal value of $\delta^{c_k}_{jl}$ from (8).

B. Optimal assignment when attackers split at $t > 0$

In reaction to the defenders' attempt to herd the attackers, the attackers may split into new smaller swarms and/or scatter as individual attackers. We continuously track the radii of the clusters and run the DBSCAN algorithm only when at some instant $t = t_{se}$ the connectivity constraint is violated by the swarms of attackers $\mathcal{A}_{c_k}(t_{se})$ for some $k \in I_{ac}(t_{se})$ i.e., when the radius $\rho_{ac_k}(t_{se})$ of the swarm of attackers $\mathcal{A}_{c_k}(t_{se})$ exceeds the value $\bar{\rho}_{ac_k}(t_{se}) = \frac{\bar{R}_{sb}}{2} \cot\left(\frac{\pi}{\mathscr{R}_d(N_a)}\right) \frac{|\mathcal{A}_{c_k}(t_{se})| - 1}{N_a - 1}$. The connectivity constraint violation is termed as split event in this paper. The split event is formally defined as:

Definition 4 (Split event)**.** *An instant t_{se} when for any swarm $\mathcal{A}_{c_k}(t_{se})$, $k \in I_{ac}(t_{se})$, the radius of the swarm of attackers $\mathcal{A}_{c_k}(t_{se})$ defined as*

[2]This is not a conservative assumption because if a defender has non-zero speed, one can apply acceleration opposite to its velocity to make the speed zero and assume the initial position for that defender to be the position at which this speed will become zero.

$\rho_{ac_k}(t_{se}) = \max_{i \in A_{c_k}(t_{se})} \|\mathbf{r}_{ai}(t_{se}) - \mathbf{r}_{ac_k}(t_{se})\|$ *exceeds the value* $\bar{\rho}_{ac_k}(t_{se})$.

We also make the following assumption regarding the splitting behavior of the attackers.

Assumption 4. *Once a swarm of attackers splits, its member attackers never rejoin each other, i.e., for all* $i \in I_a$*, if* $\exists\, t > 0$ *such that* $\mathcal{A}_i \notin \mathcal{A}_{c_k}(t)$ *for any* $k \in I_{ac}(t)$ *then* $\mathcal{A}_i \notin \mathcal{A}_{c_k}(t')$ *for all* $t \leq t'$.

The splitting behavior of the attackers requires reassignment of the defenders, that were supposed to herd the given swarm of the attackers that just split, to the newly available interception or herding tasks. Next, we describe a mixed-integer quadratically constrained quadratic program (MIQCQP) to solve this assignment problem.

1) Decentralized optimal assignment using MIQCQP: When a swarm of attackers $\mathcal{A}_{c_k}$ splits into smaller swarms at $t = t_{se}$. The newly identified swarms of the attackers by the DBSCAN algorithm are assigned new indices. Namely, one of the swarm is assigned the index k, i.e. the index of the parent swarm $\mathcal{A}_{c_k}$ and the rest swarms are assigned integers greater than $N_{ac}(t_{se}^-)$ as their indices, where t_{se}^- denotes the instant immediately before $t = t_{se}$. Let $A_c^{(k)}(t_{se})$ denote the indices of the clusters of the attackers that are newly formed out of the parent cluster $\mathcal{A}_{c_k}(t_{se}^-)$, when the cluster $\mathcal{A}_{c_k}$ splits at $t = t_{se}$, as identified by the DBSCAN algorithm. $\mathcal{A}_{uc}^{(k)}(t_{se})$ is the set of unclustered attackers separated from the original cluster $\mathcal{A}_{c_k}(t_{se})$ after the original cluster has split. We aim to assign the defenders in $\mathcal{D}_{c_k}(t_{se}^-)$, that are already connected via Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$ and were tasked to herd the original cluster $\mathcal{A}_{c_k}(t_{se}^-)$, to either intercept the unclustered attackers separated from the original cluster $\mathcal{A}_{c_k}(t_{se}^-)$ or herd the smaller clusters formed by the attackers in the original swarm $\mathcal{A}_{c_k}(t_{se}^-)$ after splitting. Herding the smaller swarms of the attackers still requires the sub-teams of the defenders to stay connected via Open-StringNets while the defenders assigned to intercept the unclustered attackers will now disconnect themselves from the rest of the Open-StrigNet. In [23], we solved a connectivity constrained generalized assignment problem (C2GAP) to assign connected sub-teams of the defenders to herd the newly formed sub-swarms of the attackers after the original attacking swarm splits. In contrast to that, the current assignment problem is more complex due to the requirement of assigning some individual defenders, who shall disconnect themselves from the rest of the Open-StringNet, to intercept the unclusterd attackers.

Let $\delta_{jk'}^{herd}(t_{se})$ be the binary decision variable at time $t = t_{se}$ that takes value 1 if the defender $\mathcal{D}_j$ is assigned to herd the swarm $\mathcal{A}_{c_{k'}}(t_{se})$ and 0 otherwise. We formulate the MIQCQP in (10) to assign the defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$ to herd the newly formed swarms of the attackers, $\mathcal{A}_{c_{k'}}(t_{se})$, for all $k' \in A_c^{(k)}(t_{se})$, and the unclustered attackers $\mathcal{A}_{uc}^{(k)}(t_{se})$. In (10), $\boldsymbol{\delta}^{(k)}(t_{se}) \in \{0,1\}^{N_{\boldsymbol{\delta}^{(k)}}(t_{se})}$ is the binary decision vector defined as $\boldsymbol{\delta}^{(k)}(t_{se}) = [[\delta_{jk'}^{herd}(t_{se}) | k' \in A_c^{(k)}(t_{se}), j \in D_{c_k}(t_{se}^-)], [\delta_{ji}^{int}(t_{se}) | i \in A_{uc}^{(k)}(t_{se}), j \in D_{c_k}(t_{se}^-)]]^T$, where $N_{\boldsymbol{\delta}^{(k)}}(t_{se}) = |\mathcal{D}_{c_k}(t_{se}^-)| \left(|A_c^{(k)}(t_{se})| + |A_{uc}^{(k)}(t_{se})|\right)$; $I'_{dc_k} = \{1, 2, ..., |\mathcal{D}_{c_k}| - 1\}$; and $\beta_k^-(l) = \beta_{c_k}(t_{se}^-, l)$.

The optimization cost in (10) is the sum of distances of the defenders from the centers of the attackers' swarms to which they are assigned, the times to capture required by the defenders to capture the unclustered attacker that are assigned to them, and the collision costs incurred by the defenders that are assigned interception task. This ensures that the collective effort needed by all the defenders is minimized when enclosing the swarms of the attackers and that the unclustered attackers are captured as quickly as possible while minimizing any possible collisions among the fast moving defenders that are assigned the interception task. The constraints (10b) ensure that each of the defenders in $\mathcal{D}_{c_k}(t_{se}^-)$ is assigned either to exactly one unclustered attacker or to exactly one swarm of the attackers. The capacity constraints (10c) ensure that for all $k' \in A_c^{(k)}(t_{se})$, the swarm $\mathcal{A}_{c_{k'}}(t_{se})$ has exactly $\mathscr{R}_d(|\mathcal{A}_{c_{k'}}(t_{se})|)$ defenders assigned to it. The constraints (10d) ensure that each unclustered attacker in $\mathcal{A}_{uc}^{(k)}(t_{se})$ has exactly one of the terminal defenders assigned to it. The quadratic constraints (10e) ensure that all the defenders assigned to swarm $\mathcal{A}_{c_{k'}}(t_{se})$ are connected together with an underlying Open-StringNet for all $k' \in A_c^{(k)}$ and the constraint (10f) ensures that all the $|D_{c_k}(t_{se}^-)|$ defenders are assigned to the attackers' swarms and the unclustered attackers.

The aforementioned MIQCQP (10) is solved by the lead defender in $\mathcal{D}_{c_k}(t_{se}^-)$, where the lead defender is identified to be the one in the middle of the Open-StringNet formation, i.e., the defender $\mathcal{D}_{\beta_k(t_{se}^-, l_i)}$ where $l_i = \lfloor \frac{|\mathcal{D}_{c_k}(t_{se}^-)|}{2} \rfloor$, for all k for which the $\mathcal{A}_{c_k}$ have split. This helps the defenders find the Defender-to-Attacker-Swarm assignment quickly, and without having to consider all the agents in the assignment formulation, i.e., in a decentralized way.

The aforementioned MIQCQP (10) can be solved using a MIP solver Gurobi [39]. After solving (10), one can find the mapping $\beta_{c_{k'}}(t, \cdot)$, for all $k' \in A_c^{(k)}(t_{se})$, as follows:

$$\beta_{c_{k'}}(t, l) = \beta_{c_k}^-(l_0 + l), \quad \forall t \in [t_{se} + t_{comp},\, t_{se}^{next}], \quad (11)$$

where l_0 is the smallest integer for which $\delta_{\beta_{c_k}^-(l_0+1)k}(t_{se}) = 1$; t_{comp} is the computation time to solve (10); and t_{se}^{next} is an unknown future time at which a split happens. In other words, the assignment obtained using the states at t_{se} continues to be a valid assignment until the next split event happens at some unknown time t_{se}^{next} in the future. The worst-case time complexity of the MIQCQP in (10) is:

$$C_M^{comp}(t_{se}, k) = O(2^{N_{\boldsymbol{\delta}^{(k)}}(t_{se})}) \quad (12)$$

where $N_{\boldsymbol{\delta}^{(k)}}(t_{se}) = |\mathcal{D}_{c_k}(t_{se}^-)| \left(|A_c^{(k)}(t_{se})| + |A_{uc}^{(k)}(t_{se})|\right)$.

C. Suboptimal assignment when attackers split at $t > 0$

1) Assignment using reduced-size MIQCQP (rs-MIQCQP): The worst-case complexity $C_M^{comp}(t_{se}, k)$

$$\boldsymbol{\delta}^{(k)*}(t_{se}) = \arg\min_{\boldsymbol{\delta}^{(k)}(t_{se})} \sum_{k' \in A_c^{(k)}(t_{se})} \sum_{j \in D_{c_k}(t_{se}^-)} \left\| \mathbf{r}_{ac_{k'}}(t_{se}) - \mathbf{r}_{dj}(t_{se}) \right\| \delta_{jk'}^{herd}(t_{se}) + \sum_{i \in A_{uc}^{(k)}(t_{se})} \sum_{j \in D_{c_k}(t_{se}^-)} C_d^{int}(\mathbf{X}_{dj}^{ai}) \delta_{ji}^{int}(t_{se})$$
$$+ \sum_{i,i' \in A_{uc}^{(k)}(t_{se})} \sum_{j,j' \in D_{c_k}(t_{se}^-)} C_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'}) \delta_{ji}^{int}(t_{se}) \delta_{j'i'}^{int}(t_{se}) \tag{10a}$$

$$\text{Subject to } \sum_{k' \in A_c^{(k)}(t_{se})} \delta_{jk'}^{herd}(t_{se}) + \sum_{i \in A_{uc}^{(k)}(t_{se})} \delta_{ji}^{int}(t_{se}) = 1, \quad \forall j \in D_{c_k}(t_{se}^-); \tag{10b}$$

$$\sum_{j \in D_{c_k}(t_{se}^-)} \delta_{jk'}^{herd}(t_{se}) = \mathscr{R}_d(|\mathcal{A}_{c_{k'}}(t_{se})|), \quad \forall k' \in A_c^{(k)}(t_{se}); \tag{10c}$$

$$\sum_{j \in D_{c_k}(t_{se}^-)} \delta_{ji}^{int}(t_{se}) = 1, \quad \forall i \in A_{uc}^{(k)}(t_{se}); \tag{10d}$$

$$\sum_{l \in I'_{dc_k}} \delta_{\beta_k^-(l)k'}^{herd}(t_{se}) \delta_{\beta_k^-(l+1)k'}^{herd}(t_{se}) \geq \mathscr{R}_d(|\mathcal{A}_{c_{k'}}(t_{se})|) - 1, \quad \forall k' \in A_c^{(k)}(t_{se}); \tag{10e}$$

$$\sum_{j \in D_{c_k}(t_{se}^-)} \left(\sum_{k' \in A_c^{(k)}(t_{se})} \delta_{jk'}^{herd}(t_{se}) + \sum_{i \in A_{uc}^{(k)}(t_{se})} \delta_{ji}^{int}(t_{se}) \right) = |D_{c_k}(t_{se}^-)|; \tag{10f}$$

$$\delta_{jk'}^{herd}(t_{se}),\ \delta_{ji}^{int}(t_{se}) \in \{0,1\}, \quad \forall j \in D_{c_k}(t_{se}^-),\ k' \in A_c^{(k)}(t_{se}),\ i \in A_{uc}^{(k)}(t_{se}); \tag{10g}$$

of the MIQCQP in (10) can be reduced further under certain assumption on the behavior of the attackers. Let us first define a conical envelope around the center of a swarm.

Definition 5 (Conical Envelope)**.** *A conical envelope* $E_{con}(\mathbf{r}_0, \psi)$*, centered at* $\mathbf{r}_0 = [x_0, y_0]^T$ *is defined as* $E_{con}(\mathbf{r}_0, \psi) = \{\{(x,y) \in \mathbb{R}^2 | y - y_0 - m_1(x - x_0) > 0\} \cap \{(x,y) \in \mathbb{R}^2 | y - y_0 - m_2(x - x_0) < 0\}\} \cup \{\{(x,y) \in \mathbb{R}^2 | y - y_0 - m_1(x - x_0) < 0\} \cap \{(x,y) \in \mathbb{R}^2 | y - y_0 - m_2(x - x_0) > 0\}\}$*, where* $m_1 = \tan\left(\tan^{-1}(\frac{y_0 - y_p}{x_0 - x_p}) - \frac{\pi}{2} - \psi\right)$ *and* $m_2 = \tan\left(\tan^{-1}(\frac{y_0 - y_p}{x_0 - x_p}) - \frac{\pi}{2} + \psi\right)$.

Assumption 5. *A swarm of the attackers* $\mathcal{A}_{c_k}$*, for any* k*, splits at* $t = t_{se}$*, such that all the unclustered attackers (swarms with less than 3 attackers) are the farthest from the center of the original swarm* $\mathcal{A}_{c_k}(t_{se}^-)$ *and their centers lie within the conical envelope* $E_{con}(\mathbf{r}_{ac_k}(t_{se}^-), \frac{\pi}{4})$*, i.e.,* $\forall i \in A_{uc}^{(k)}(t_{se})$, $\|\mathbf{r}_{ai}(t_{se}) - \mathbf{r}_{ac_k}(t_{se}^-)\| > \max_{k' \in A_c^{(k)}(t_{se})} \left\|\mathbf{r}_{ac_{k'}}(t_{se}) - \mathbf{r}_{ac_k}(t_{se}^-)\right\|$ *and* $\mathbf{r}_{ai}(t_{se}) \in E_{con}(\mathbf{r}_{ac_k}(t_{se}^-), \frac{\pi}{4})$ *(gray shaded region in Fig. 3).*

Assumption 5 implies that the unclustered attackers aim to spread in the direction transverse to the direction toward the protected area because of the presence of the defenders in front of them in order to maximize their chances of not getting captured by the defenders and reaching the protected area. Under Assumption 5, we can assign only the defenders from either end of the Open-StringNet to intercept the unclustered attackers while assign the defenders in the central part of the Open-StringNet to herd the newly formed clusters of the attackers.

Let $\mathcal{D}_{c_k}^l(t_{se}^-) = \{\mathcal{D}_j | j \in D_{c_k}^l(t_{se}^-)\}$ be the group of $|A_{uc}^{(k)}(t_{se})|$ defenders at the left end of the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$, where $D_{c_k}^l(t_{se}^-) = \{\beta_{c_k}^-(1), \beta_{c_k}^-(2), ..., \beta_{c_k}^-(|A_{uc}^{(k)}(t_{se})|)\}$. Here the left end of the Open-StringNet formation refers to the end approached first when one rotates anti-clockwise standing at the center $\mathbf{r}_{df_k}$ and starting when facing in the direction ϕ_k of the formation (see Fig. 3). Similarly, let $\mathcal{D}_{c_k}^r(t_{se}^-) = \{\mathcal{D}_j | j \in D_{c_k}^r(t_{se}^-)\}$ be the group of $|A_{uc}^{(k)}(t_{se})|$ defenders at the right end of the Open-StringNet formation $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$, where $D_{c_k}^r(t_{se}^-) = \{\beta_{c_k}^-(|D_{c_k}(t_{se}^-)| - |A_{uc}^{(k)}| + 1), \beta_{c_k}^-(|D_{c_k}(t_{se}^-)| - |A_{uc}^{(k)}| + 2), ..., \beta_{c_k}^-(|D_{c_k}(t_{se}^-)|)\}$ (see Fig. 3). Let us call $\mathcal{D}_{c_k}^t(t_{se}^-) = \{\mathcal{D}_j | j \in D_{c_k}^l(t_{se}^-) \cup D_{c_k}^r(t_{se}^-)\}$ as the group of terminal defenders of the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$. We denote by $\mathcal{D}_{c_k}^c(t_{se}^-) = \{\mathcal{D}_j | j \in D_{c_k}^c(t_{se}^-)\}$ the central defenders, the group of the defenders excluding the terminal defenders $\mathcal{D}_{c_k}^t(t_{se}^-)$, where $D_{c_k}^c(t_{se}^-) = D_{c_k}(t_{se}^-) \backslash D_{c_k}^t(t_{se}^-)$.

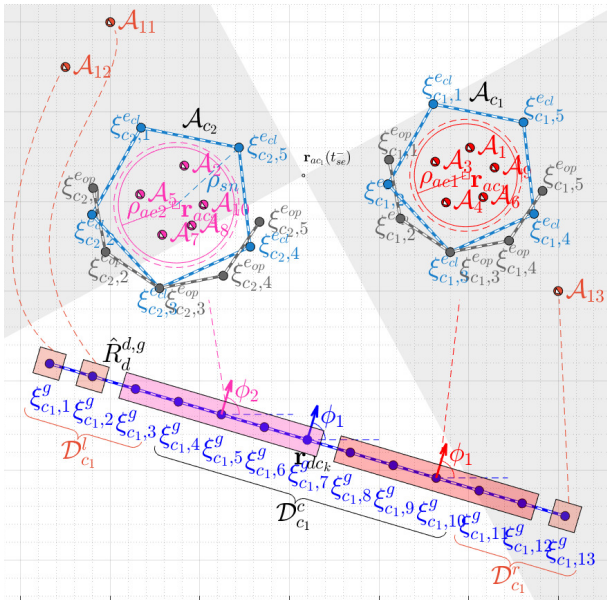


Figure 3: Assignment of the defenders after the attackers split using rs-MIQCQP

Next, we develop a reduced-size MIQCQP, in which only the terminal defenders $\mathcal{D}_{c_k}^t(t_{se}^-)$ are assigned the interception task, in (13). In (13) the length of the decision vector $\boldsymbol{\delta}_{rs}^{(k)*}(t_{se}) = [[\delta_{jk'}^{herd}(t_{se}) | k' \in A_c^{(k)}(t_{se}),\ j \in D_{c_k}(t_{se}^-)], [\delta_{ji}^{int}(t_{se}) | i \in A_{uc}^{(k)}(t_{se}),\ j \in$

$$\boldsymbol{\delta}_{rs}^{(k)*}(t_{se}) = \underset{\boldsymbol{\delta}_{rs}^{(k)}(t_{se})}{\arg\min} \sum_{k'\in A_c^{(k)}(t_{se})} \sum_{j\in D_{c_k}(t_{se}^-)} \left\|\mathbf{r}_{ac_{k'}}(t_{se}) - \mathbf{r}_{dj}(t_{se})\right\| \delta_{jk'}^{herd}(t_{se}) + \sum_{i\in A_{uc}^{(k)}(t_{se})} \sum_{j\in D_{c_k}^t(t_{se}^-)} C_d^{int}(\mathbf{X}_{dj}^{ai})\delta_{ji}^{int}(t_{se})$$
$$+ \sum_{i,i'\in A_{uc}^{(k)}(t_{se})} \sum_{j,j'\in D_{c_k}^t(t_{se}^-)} C_d^{col}(\mathbf{X}_{dj}^{ai}, \mathbf{X}_{dj'}^{ai'})\delta_{ji}^{int}(t_{se})\delta_{j'i'}^{int}(t_{se}) \tag{13a}$$

$$\text{Subject to } \textstyle\sum_{k'\in A_c^{(k)}(t_{se})} \delta_{jk'}^{herd}(t_{se}) + \sum_{i\in A_{uc}^{(k)}(t_{se})} \delta_{ji}^{int}(t_{se}) = 1, \quad \forall j \in D_{c_k}^t(t_{se}^-); \tag{13b}$$

$$\textstyle\sum_{k'\in A_c^{(k)}(t_{se})} \delta_{jk'}^{herd}(t_{se}) = 1, \quad \forall j \in D_{c_k}^c(t_{se}); \tag{13c}$$

$$\textstyle\sum_{j\in D_{c_k}(t_{se}^-)} \delta_{jk'}^{herd}(t_{se}) = \mathscr{R}_d(|\mathcal{A}_{c_{k'}}(t_{se})|), \quad \forall k' \in A_c^{(k)}(t_{se}); \tag{13d}$$

$$\textstyle\sum_{j\in D_{c_k}^t(t_{se}^-)} \delta_{ji}^{int}(t_{se}) = 1, \quad \forall i \in A_{uc}^{(k)}(t_{se}); \tag{13e}$$

$$\textstyle\sum_{l\in I'_{dc_k}} \delta_{\beta_k^-(l)k'}^{herd}(t_{se})\delta_{\beta_k^-(l+1)k'}^{herd}(t_{se}) \geq \mathscr{R}_d(|\mathcal{A}_{c_{k'}}(t_{se})|) - 1, \quad \forall k' \in A_c^{(k)}(t_{se}); \tag{13f}$$

$$\textstyle\sum_{j\in D_{c_k}^c(t_{se}^-)} \sum_{k'\in A_c^{(k)}(t_{se})} \delta_{jk'}^{herd}(t_{se}) + \sum_{j\in D_{c_k}^t(t_{se}^-)} \sum_{i\in A_{uc}^{(k)}(t_{se})} \delta_{ji}^{int}(t_{se}) = |D_{c_k}(t_{se}^-)|; \tag{13g}$$

$$\delta_{jk'}^{herd}(t_{se}),\ \delta_{ji}^{int}(t_{se}) \in \{0,1\}, \quad \forall j \in D_{c_k}(t_{se}^-),\ k' \in A_c^{(k)}(t_{se}),\ i \in A_{uc}^{(k)}(t_{se}); \tag{13h}$$

$D_{c_k}(t_{se}^-)]]^T$ is $N_{\boldsymbol{\delta}_{rs}^{(k)}}(t_{se}) = |\mathcal{D}_{c_k}(t_{se}^-)||A_c^{(k)}(t_{se})| + \min(2|A_{uc}^{(k)}(t_{se})|, |\mathcal{D}_{c_k}(t_{se}^-)|)|A_{uc}^{(k)}(t_{se})|$. We have the following result about the computation cost of (13).

Lemma 1. *The worst-case computational cost of* (13), $C_{rsM}^{comp}(t_{se}, k)$*, satisfies:*

$$C_{rsM}^{comp}(t_{se}, k) = O(2^{N_{\boldsymbol{\delta}_{rs}^{(k)}}(t_{se})}) \leq C_M^{comp}(t_{se}, k). \tag{14}$$

Furthermore, if the number of unclustered attackers is less than half of the total number of attackers in the original cluster, i.e., $|A_{uc}^{(k)}(t_{se})| < \frac{|A_{c_k}(t_{se}^-)|}{2}$*, then* $C_{rsM}^{comp}(t_{se}, k) < C_M^{comp}(t_{se}, k)$.

Figure 3 shows an instance of the assignment of the defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_1}(t_{se}^-))$ at some time $t = t_{se}$, where $D_{c_1}(t_{se}^-) = \{1, 2, 3, ..., 13\}$, to the newly formed clusters $\mathcal{A}_{c_1}(t_{se}) = \{\mathcal{A}_1, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_6, \mathcal{A}_9\}$, $\mathcal{A}_{c_2}(t_{se}) = \{\mathcal{A}_2, \mathcal{A}_5, \mathcal{A}_7, \mathcal{A}_8, \mathcal{A}_{10}\}$ and the unclustrered attackers $\mathcal{A}_{uc}^{(1)}(t_{se}) = \{\mathcal{A}_{11}, \mathcal{A}_{12}, \mathcal{A}_{13}\}$. After solving the rs-MIQCQP (13), as shown in Fig. 3, defenders $\mathcal{D}_{\beta_1^-(1)}$, $\mathcal{D}_{\beta_1^-(2)}$ and $\mathcal{D}_{\beta_1^-(13)}$ are assigned to the unclustered attackers $\mathcal{A}_{12}$, $\mathcal{A}_{11}$, $\mathcal{A}_{13}$, respectively, so that these attackers can be intercepted as soon as possible. The connected sub-teams $\{\mathcal{D}_{\beta_1^-(8)}, \mathcal{D}_{\beta_1^-(9)}, \mathcal{D}_{\beta_1^-(10)}, \mathcal{D}_{\beta_1^-(11)}, \mathcal{D}_{\beta_1^-(12)}\}$ and $\{\mathcal{D}_{\beta_1^-(3)}, \mathcal{D}_{\beta_1^-(4)}, \mathcal{D}_{\beta_1^-(5)}, \mathcal{D}_{\beta_1^-(6)}, \mathcal{D}_{\beta_1^-(7)}\}$ are assigned to the newly formed swarms of the attackers $\mathcal{A}_{c_1}(t_{se})$ and $\mathcal{A}_{c_2}(t_{se})$, respectively.

2) Hierarchical approach to assignment (a heuristic): Finding the optimal assignment of the defenders for interception and herding tasks by solving the MIQCQPs (10) and (13) may not be real-time implementable for a large number of agents (> 100). In this subsection, we develop a computationally efficient hierarchical approach to find the defender-to-attacker-swarm assignment under Assumption 5. The idea is to split a large dimensional assignment problem into smaller, low-dimensional assignment problems that can be solved optimally and quickly.

Let $\mathscr{A}_k(t_{se})$ be a data structure that stores information about the attackers in $\mathcal{A}_{c_k}(t_{se}^-)$ and has data fields: $\mathscr{A}_k(t_{se}).\mathbf{r}_{ac} = [\mathbf{r}_{ac_{k'}}|k' \in A_c^{(k)}(t_{se})]$, centers of the newly formed attackers' swarms after separating from the original swarm $\mathcal{A}_{c_k}(t_{se}^-)$; $\mathscr{A}_k(t_{se}).\mathbf{n}_{ac} = [|\mathcal{A}_{c_{k'}}(t_{se})||k' \in A_c^{(k)}(t_{se})]$, numbers of the attackers in each swarm; $\mathscr{A}_k(t_{se}).N_{ac} = |A_c^{(k)}(t_{se})|$, total number of attackers' clusters formed from $\mathcal{A}_k(t_{se}^-)$; $\mathscr{A}_k(t_{se}).\mathbf{r}_{uc} = [\mathbf{r}_{ai}|i \in A_{uc}^{(k)}(t_{se})]$ current states of the unclustered attackers in $\mathcal{A}_{uc}^{(k)}(t_{se})$; $\mathscr{A}_k(t_{se}).N_{uc}$, total number of unclustered attackers; $\mathscr{A}_k(t_{se}).N_a = |\mathcal{A}_{c_k}(t_{se}^-)|$, total number of attackers $\mathcal{A}_{c_k}(t_{se}^-)$. Similarly, $\mathscr{D}_k(t_{se})$ is a data structure that stores the information of the defenders on the original Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$ with data fields: $\mathscr{D}_k(t_{se}).\mathbf{r}_d = [\mathbf{r}_{dj}|j \in D_{c_k}(t_{se}^-)]$, positions of the defenders on $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$; and $\mathscr{D}_k(t_{se}).\beta = \beta_{c_k}(t_{se}^-)$, the original assignment mapping of the defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$.

Algorithm 2 provides the steps to solve the assignment problem quickly by hierarchically reducing the original big assignment problem into smaller ones.

In Algorithm 2, the function `splitUnclustAtt` $(\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se}))$ splits the unclustered attackers $\mathcal{A}_{uc}^{(k)}(t_{se})$ into two groups: left group $\mathcal{A}_{uc}^{(k),l}(t_{se})$ and right group $\mathcal{A}_{uc}^{(k),r}(t_{se})$. The normal bisector of the line segment joining the positions $\mathbf{r}_{d\beta_{c_k}^-(1)}(t_{se})$ and $\mathbf{r}_{d\beta_{c_k}^-(|D_{c_k}|)}(t_{se})$ acts as separating hyperplane for the groups $\mathcal{A}_{uc}^{(k),l}(t_{se})$ and $\mathcal{A}_{uc}^{(k),r}(t_{se})$. The unclustered attackers that lie in the half-plane containing the left side of Open-StringNet and the normal bisector itself are part of the left group $\mathcal{A}_{uc}^{(k),l}(t_{se})$ and the rest unclustered attackers in $\mathcal{A}_{uc}^{(k)}(t_{se})$ are part of the right group $\mathcal{A}_{uc}^{(k),r}(t_{se})$ (see Fig. 4). The function `splitUnclustAtt` also outputs $\mathcal{D}_{uc}^{(k),l}(t_{se})$, the leftmost $|\mathcal{A}_{uc}^{(k),l}(t_{se})|$ defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$; and $\mathcal{D}_{uc}^{(k),r}(t_{se})$, the rightmost $|\mathcal{A}_{uc}^{(k),r}(t_{se})|$ defenders on the Open-StringNet $\mathcal{G}_{sn}^{op}(D_{c_k}(t_{se}^-))$ (see Fig. 4). The function `CADAA` $(\mathcal{A}_{uc}^{(k),l}(t_{se}), \mathcal{D}_{uc}^{(k),l}(t_{se}))$ assigns the defenders in $\mathcal{D}_{uc}^{(k),l}(t_{se})$ to intercept the

Algorithm 2: Defender-to-Attacker-Swarm Assignment (DASA)

Input: $\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se})$

1 $[\mathcal{A}_{uc}^{(k),l}(t_{se}),\ \mathcal{D}_{uc}^{(k),l}(t_{se}), \mathcal{A}_{uc}^{(k),r}(t_{se}), \mathcal{D}_{uc}^{(k),r}(t_{se})] =$

2 `splitUnclustAtt` $(\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se}))$;

3 $\beta_{uc}^{(k),l}$= `CADAA` $(\mathcal{A}_{uc}^{(k),l}(t_{se}), \mathcal{D}_{uc}^{(k),l}(t_{se}))$;

4 $\beta_{uc}^{(k),r}$= `CADAA` $(\mathcal{A}_{uc}^{(k),r}(t_{se}), \mathcal{D}_{uc}^{(k),r}(t_{se}))$;

5 $\beta_{uc}(t_{se}) \leftarrow \{\beta_{uc}(t_{se}),\ \beta_{uc}^{(k),l} \cup \beta_{uc}^{(k),r}\}$;

6 $\mathscr{D}_k(t_{se}).\mathcal{D}_{c_k} \leftarrow$
$(\mathscr{D}_k(t_{se}).\mathcal{D}_{c_k})\backslash(\mathcal{D}_{uc}^{(k),l}(t_{se}) \cup \mathcal{D}_{uc}^{(k),r}(t_{se}))$;

7 $\beta_c(t_{se}) \leftarrow$
$\{\beta_c(t_{se}),$ `assignHierarchical`$(\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se}))\}$;

8 **return** $\beta_{uc}(t_{se}), \beta_c(t_{se})$;

---

9 **Function** `assignHierarchical`$(\mathscr{A}_k, \mathscr{D}_k)$**:**

10 **if** $\mathscr{A}_k.N_{ac} > \underline{N}_{ac}$ **then**

11 $[\mathscr{A}_k^l, \mathscr{D}_k^l, \mathscr{A}_k^r, \mathscr{D}_k^r] =$

12 `splitClustersEqual` $(\mathscr{A}_k, \mathscr{D}_k)$;

13 **for** $\iota \in \{l, r\}$ **do**

14 **if** $\mathscr{A}_k^\iota.N_{ac} > \underline{N}_{ac}$ **then**

15 $\beta_{c_k}^\iota =$ `assignHierarchical` $(\mathscr{A}_k^\iota, \mathscr{D}_k^\iota)$;

16 **else**

17 $\beta_{c_k}^\iota =$ `assignMIQCQP` $(\mathscr{A}_k^\iota, \mathscr{D}_k^\iota)$;

18 $\beta_c = \{\beta_{c_k}^l, \beta_{c_k}^r\}$;

19 **else**

20 β_c=`assignMIQCQP` $(\mathscr{A}_k(t_{se}), \mathscr{D}_k)$;

21 **return** β_c;

attackers $\mathcal{A}_{uc}^{(k),l}(t_{se})$ by solving CADAA (5). Line 6 in Algorithm 2 removes the the defenders in $\mathcal{D}_{uc}^{(k),l}(t_{se})$ and $\mathcal{D}_{uc}^{(k),r}(t_{se})$, that are already assigned to intercept the unclustered attackers, from further processing. The function `assignHierarchical`$(\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se}))$ then assigns the remaining connected defenders on the Open-StringNet to the clusters of the attackers $\{\mathcal{A}_{c_{k'}}(t_{se}) | k' \in A_c^{(k)}(t_{se})\}$.

In the function `assignHierarchical`, the function `splitClustersEqual` $(\mathscr{A}_k(t_{se}), \mathscr{D}_k(t_{se}))$ splits the clusters of the attackers into two groups $\mathscr{A}_k^l(t_{se})$ and $\mathscr{A}_k^r(t_{se})$ of roughly equal number of attackers and the defenders into two groups $\mathscr{D}_k^l(t_{se})$ and $\mathscr{D}_k^r(t_{se})$. The split is performed based on the angles $\psi_{k'}$ made by relative vectors $\mathbf{r}_{ac_{k'}}(t_{se}) - \mathbf{r}_{dc_k}(t_{se}^-)$, for all $k' \in A_c^{(k)}(t_{se})$, with the vector $\mathbf{r}_{dj_t}(t_{se}^-) - \mathbf{r}_{dc_k}(t_{se}^-)$ where $\mathbf{r}_{dc_k}(t_{se}^-) = \frac{\mathbf{r}_{dj_1}(t_{se}) + \mathbf{r}_{dj_t}(t_{se})}{2}$ is the center of $\mathcal{D}_{c_k}(t_{se}^-)$, where $j_1 = \beta_{c_k}^-(1)$ and $j_t = \beta_{c_k}^-(|D_{c_k}(t_{se}^-)|)$. We first arrange these angles $\psi_{k'}$ in the descending order. The first few clusters in the arranged list with roughly half the total number of attackers become the left group $\mathscr{A}_k^l(t_{se})$ and the rest become the right group $\mathscr{A}_k^r(t_{se})$ (see Fig. 4). Similarly, the left group $\mathscr{D}_k^l(t_{se})$ is formed by the first $\mathscr{A}_k^l(t_{se}).N_a$ defenders as per the assignment $\beta_{c_k}^-$ and the rest defenders form the right group $\mathscr{D}_k^r(t_{se})$ (see Fig. 4). We assign the defenders in $\mathscr{D}_k^l(t_{se})$ only to the swarms in $\mathscr{A}_k^l(t_{se})$ and those in $\mathscr{D}_k^r(t_{se})$ only to the swarms in $\mathscr{A}_k^r(t_{se})$. By doing so we may or may not obtain an assignment that minimizes the cost in (10a) but we reduce the computation time significantly and obtain a reasonably good assignment quickly. As in the function `assignHierarchical`, the process of splitting is done recursively until the number of attackers' swarms is smaller than a pre-specified number $\underline{N}_{ac}(> 2)$. The function `assignMIQCQP` finds the defender-to-attacker-swarm assignment $\beta_c(t_{se})$ by solving (13) after setting $A_{uc}^{(k)}(t_{se})$ and $D_{c_k}^t(t_{se}^-)$ as empty sets, i.e., no assignments of the terminal defenders to the unclustered attackers as this assignment is already performed in the prior steps.

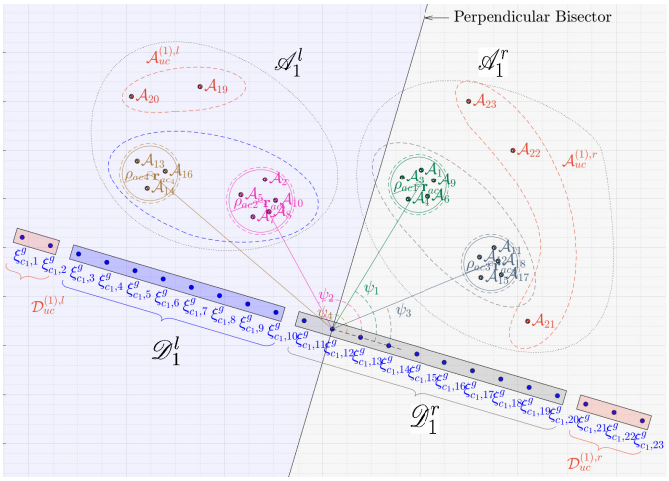


Figure 4: Grouping for the hierarchical algorithm

We have the following result about the worst-case computational cost of the hierarchical heuristic.

Lemma 2. *For a given assignment problem of assigning $|D_{c_k}(t_{se}^-)|$ defenders to $\mathscr{A}_k(t_{se}).N_a$ $(= |D_{c_k}(t_{se}^-)|)$ attackers divided into $\mathscr{A}_k(t_{se}).N_{ac}$ clusters and $\mathscr{A}_k(t_{se}).N_{uc}$ unclustered attackers with a given threshold $\underline{N}_{ac}(> 2)$, the worst-case computational cost of the hierarchical heuristic in Algorithm 2 is:*

$$C_H^{comp}(t_{se}, k) = O(2^{(\mathscr{A}_k(t_{se}).N_{uc})^2} + (N'_{rsM} - 1)2^{3\underline{N}_{ac}^2} + 2^{\underline{N}_{ac} n_{max}} + 2^{3n_{ac,k}^2}) \tag{15}$$

where $N'_{rsM} = \lfloor \frac{\mathscr{A}_k(t_{se}).N_{ac}}{\underline{N}_{ac}} \rfloor$, $n_{max} = (\mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc} - 3n_{ac,k} - 3\underline{N}_{ac}(N'_{rsM} - 1))$ and $n_{ac,k} = \mathscr{A}_k(t_{se}).N_{ac} - N'_{rsM}\underline{N}_{ac}$.

Proof: In Algorithm 2, two CADAA problems (mixed integer quadratic programs) are solved (line 3 and 4) to assign the defenders to the left and right group of unclustered attackers. Suppose the number of unclustered attackers in left and right group are $N_{uc}^l = |\mathcal{A}_{uc}^{(k),l}(t_{se})|$ and $N_{uc}^r = |\mathcal{A}_{uc}^{(k),r}(t_{se})|$, respectively.

Additionally, there are several rs-MIQCQPs that are solved in Algorithm 2 to assign defenders to the clusters of the attackers. Maximum number of the clusters in any rs-MIQCQP solved in Algorithm 2 is $\underline{N}_{ac}$. Based on the hierarchical breakdown of the original assignment

problem, the maximum number of such rs-MIQCQP's is $N'_{rsM} = \lfloor \frac{\mathscr{A}_k(t_{se}).N_{ac}}{\underline{N}_{ac}} \rfloor$. Let n_i ($\geq 3\underline{N}_{ac}$) denote the number of attackers in the $\underline{N}_{ac}$ clusters in the i^{th} rs-MIQCQP for all $i \in \{1, 2, 3, ..., N'_{rsM}\}$. Similarly, let n_0 be the number of attackers in the remaining $n_{ac,k} = \mathscr{A}_k(t_{se}).N_{ac} - N'_{rsM}\underline{N}_{ac}$ clusters considered in a separate rs-MIQCQP. We also have that equal number of defenders are to be assigned to these attackers by solving these integer programs. Then, the worst-case computational cost of solving all integer programs in Algorithm 2 is:

$$C^{comp} = O\big(\underbrace{2^{(N^l_{uc})^2} + 2^{(N^r_{uc})^2}}_{C^{comp}_{uc}} + \underbrace{2^{n_0 n_{ac,k}} + \sum_{i=1}^{N'_{rsM}} 2^{n_i \underline{N}_{ac}}}_{C^{comp}_{c}}\big) \tag{16}$$

where $N^l_{uc} + N^r_{uc} = \mathscr{A}_k(t_{se}).N_{uc}$, and $\sum_{i=0}^{N'_{rsM}} n_i = \mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc}$. Since the assignments to unclustered and clustered attackers are made separately, we will find the maximum values of C^{comp}_{uc} and C^{comp}_{c} separately. The maximum value of C^{comp}_{uc} occurs when either $N^l_{uc} = \mathscr{A}_k(t_{se}).N_{uc}$ and $N^r_{uc} = 0$ or $N^l_{uc} = 0$ and $N^r_{uc} = \mathscr{A}_k(t_{se}).N_{uc}$. We have that $n_{ac,k} \leq \underline{N}_{ac}$. Then, the maximum value of C^{comp} subject to $\sum_{i=1}^{N'_{rsM}} n_i = \mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc}$ occurs when all n_i, except one n_i for some $i \in \{1, 2, 3, ..., N'_{rsM}\}$, take their smallest values, i.e., when $n_0 = 3n_{ac,k}$, $n_i = 3\underline{N}_{ac}$ for all $i \in \{2, 3, ..., N'_{rsM}\}$ and $n_1 = n_{max} = \mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc} - 3n_{ac,k} - 3\underline{N}_{ac}(N'_{rsM} - 1)$. Hence, the worst-case computational cost of the hierarchical heuristic is $C^{comp}_H(t_{se}, k) = O(2^{(\mathscr{A}_k(t_{se}).N_{uc})^2} + (N'_{rsM} - 1)2^{3\underline{N}^2_{ac}} + 2^{\underline{N}_{ac}(\mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc} - 3n_{ac,k} - 3\underline{N}_{ac}(N'_{rsM}-1))} + 2^{3n^2_{ac,k}})$. ∎

D. Assignment when attackers' swarm does not avoid defenders

When the attackers in a given swarm $\mathcal{A}_{c_k}(t)$ do not try to avoid the defenders and instead just aim to reach the protected area, i.e., the attackers are risk-taking, then herding will not be an effective way of defense. Mathematically, this intention of swarm of attackers $\mathcal{A}_{c_k}(t)$ to not avoid defenders and simply target protected area, is characterized by the following condition.

$$\|\mathbf{r}_{ac_k} - \mathbf{r}_p\| \leq \|\mathbf{r}_{df_k}(0) - \mathbf{r}_p\| \ \& \ (\mathbf{r}_{ac_k} - \mathbf{r}_p)^T \mathbf{v}_{ac_k} < 0 \tag{17}$$

This condition implies that the center of mass of attackers in $\mathcal{A}_{c_k}(t)$ has come closer towards the protected area than the gathering center of the corresponding herding defenders in $\mathcal{D}_{c_k}(t)$ and the attackers' average velocity vector points towards the protected area. In other words, the attackers in $\mathcal{A}_{c_k}(t)$ are not necessarily moving away from the defenders and they intend to simply reach the protected area $\mathcal{P}$, i.e., the attackers are risk taking. Once swarm $\mathcal{A}_{c_k}(t)$ satisfies (17), the corresponding defenders $\mathcal{D}_{c_k}(t)$ choose to intercept all the attackers in $\mathcal{A}_{c_k}(t)$. The defenders in $\mathcal{D}_{c_k}$ are assigned to intercept the attackers in $\mathcal{A}_{c_k}$ by using CADAA similar to (5) with $A_{c_k}(t)$ and $D_{c_k}(t)$ at the place of $A_{uc}(0)$ and I_d, respectively.

E. Comparison of the assignment algorithms

In this section, we compare the computational performance of the assignment algorithms. Using the results from Lemma 1 and 2, we have the following result about the computational cost of the MIQCQP, the rs-MIQCQP and the heuristic in Algorithm 2.

Theorem 3. *Let Assumption 5 hold and $1 < \underline{N}_{ac} < N_{ac}$, then the worst-case computational costs C^{comp}_M, C^{comp}_{rsM} and C^{comp}_H of the MIQCQP, the rs-MIQCQP and the heuristic, respectively, satisfy: $C^{comp}_H(t_{se}, k) < C^{comp}_{rsM}(t_{se}, k) \leq C^{comp}_M(t_{se}, k)$.*

Proof: From Lemma 2, we have:

$$\begin{aligned}
C^{comp}_H &= C^{comp}_H(t_{se}, k) \\
&= O\big(2^{(\mathscr{A}_k(t_{se}).N_{uc})^2} + (N'_{rsM} - 1)2^{3\underline{N}^2_{ac}} \\
&\quad + 2^{\underline{N}_{ac} n_{max}} + 2^{3n^2_{ac,k}}\big) \\
&\leq O\big(2^{(\mathscr{A}_k(t_{se}).N_{uc})^2 + 3\underline{N}^2_{ac}(N'_{rsM}-1) + \underline{N}_{ac} n_{max} + 3n^2_{ac,k}}\big) \\
&\qquad (\because 2^\imath + 2^\jmath \leq 2^{\imath+\jmath}, \forall \imath, \jmath \geq 1) \\
&\leq O\big(2^{\left((\mathscr{A}_k(t_{se}).N_{uc})^2 + \underline{N}_{ac}(\mathscr{A}_k(t_{se}).N_a - \mathscr{A}_k(t_{se}).N_{uc})\right)} \times \\
&\quad 2^{-3\underline{N}_{ac} n_{ac,k} + 3n^2_{ac,k}}\big) \\
&\leq O\big(2^{\left((\mathscr{A}_k(t_{se}).N_{uc})^2 + \underline{N}_{ac}(\mathscr{A}_k(t_{se}).N_a\right)}\big) \\
&\qquad (\because n_{ac,k} \leq \underline{N}_{ac}) \\
&< O\big(2^{\min(2\mathscr{A}_k(t_{se}).N_{uc}, |\mathcal{D}_{c_k}(t^-_{se})|)\mathscr{A}_k(t_{se}).N_{uc}} \times \\
&\quad 2^{|\mathcal{D}_{c_k}(t^-_{se})|\mathscr{A}_k(t_{se}).N_{ac}}\big) \\
&\qquad (\because 1 < \underline{N}_{ac} < N_{ac}) \\
&= C^{comp}_{rsM}(t_{se}, k)
\end{aligned} \tag{18}$$

Using (18) and the result from Lemma 1, we can establish: $C^{comp}_H(t_{se}, k) < C^{comp}_{rsM}(t_{se}, k) \leq C^{comp}_M(t_{se}, k)$. ∎

Next, we analyze the average computational performance of the assignment algorithms by numerically evaluating random assignment scenarios on a computer with 16 core Intel-i7 processor and 64 GB RAM using MATLAB. The computation time for random initializations of the players for different numbers of clusters of the attackers and different numbers of the unclustered attackers is shown in Figure 5(a), and that for different numbers of attackers is shown in Figure 5(b). Each data point in Fig. 5 is obtained by taking average of the computational costs for 30 random sets of initial conditions of the players for each of the possible configurations of the clusters for the given number of clusters and the total number of agents. As one can observe, the computation time for MIQCQP increases with increase in total number of attackers as well as number of unclustered attackers. Furthermore, even for $N_a = 30$ and $N_{uc} = 8$, the MIQCQP in 13 takes around 25 s, which is not real-time implementable. Similarly, we show the computation times for the rs-MIQCQP and the hierarchical heuristic in Figure 6 and 7, respectively. As one can observe, the computational time for the respective scenarios for the rs-MIQCQP is significantly smaller than that for the MIQCQP, but rs-MIQCQP could still be too slow for a real-time operation. The heuristic has even smaller computation time than the rs-MIQCQP and thus

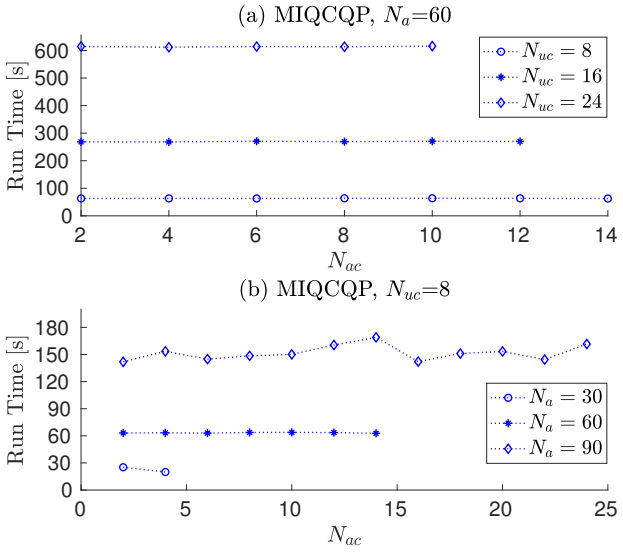


Figure 5: Computation time for MIQCQP in (10)

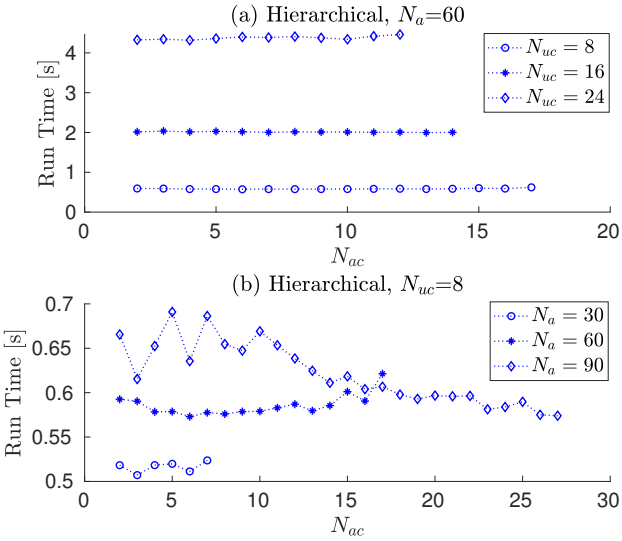


Figure 7: Computation time for Hierarchical Approach in Algorithm 2

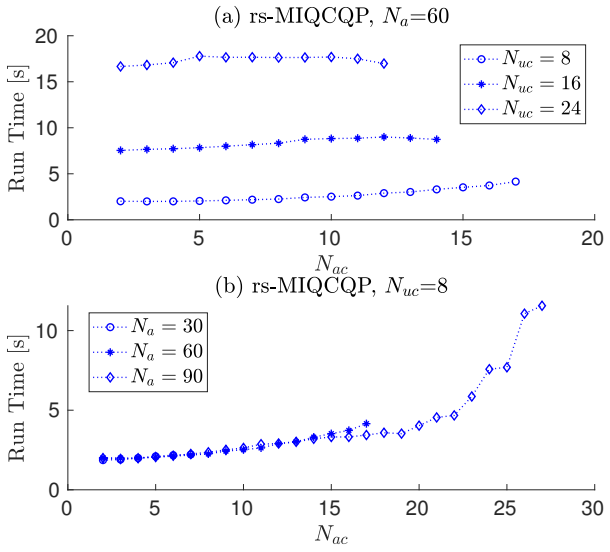


Figure 6: Computation time for rs-MIQCQP in (13)

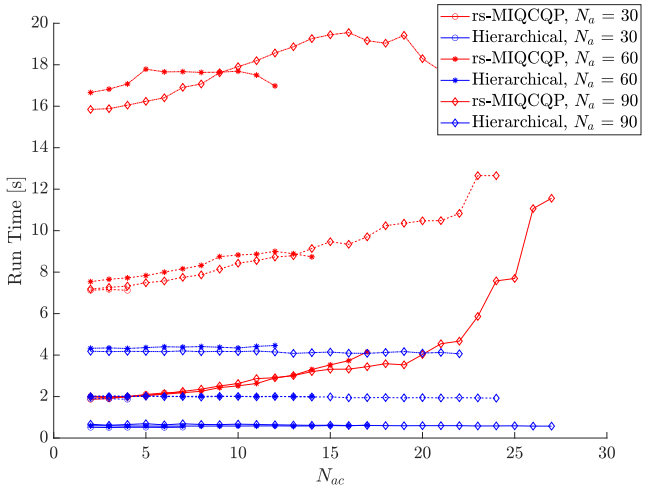


Figure 8: Comparison of computation times of the rs-MIQCQP and the hierarchical heuristic (The line types solid (-), dash (- -), and dash-dot (-.) correspond to the cases with $N_{uc} = 8$, $N_{uc} = 16$, $N_{uc} = 24$ respectively.)

more suitable for real-time operation, see the Figure 8 for better comparison.

We also compare the resulting cost of the heuristic, $cost_H$, against the optimal cost, $cost_{rsM}$, obtained by solving the rs-MIQCQP by calculating the percentage error $\%E = \frac{100|cost_{rsM} - cost_H|}{cost_{rsM}}$. As one can observe in Fig.9 the percentage error $\%E$ is below 4% for all the evaluated cases. This means that the proposed heuristic provides an assignment solution that is very close to the one obtained by rs-MIQCQP within a fraction of the time taken by rs-MIQCQP. The heuristic algorithm can be run at around 2-5 Hz for problems with up to 60 attackers and up to 24 individual risk taking attackers. The analysis providing theoretical guarantees on the cost of the heuristic is left open for future research.

F. Control augmentation for inter-defender collision avoidance

The intercepting defenders need to avoid collisions with other intercepting as well as the herding defenders for their own safety. Each intercepting defender $\mathcal{D}_j$, for all $j \in D_{uc}(t)$, employs an exponential CBF (ECBF) [40], [41] based control augmentation to avoid collisions with other defenders such that their time-optimal control action corresponding to their assigned attacker is minimally augmented. This ECBF based control considers the Open-StringNets and Close-StringNets formed by the sub-teams of the herding defenders as big individual agents with their corresponding formation radii that the individual intercepting defenders need to avoid.

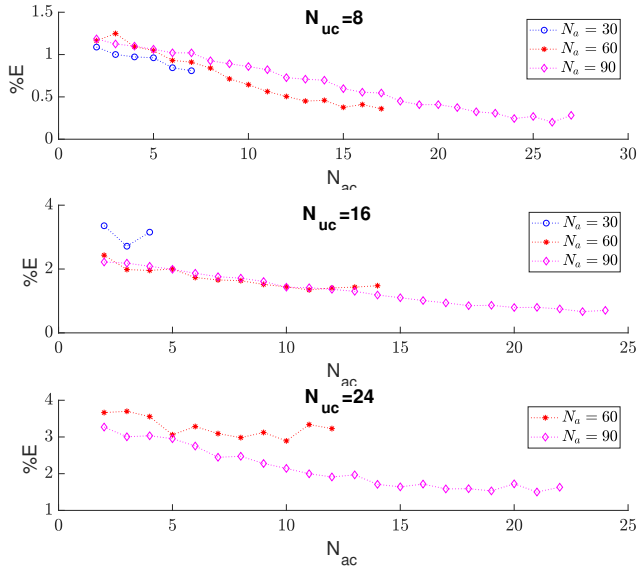


Figure 9: % Error in the costs of the rs-MIQCQP and the hierarchical heuristic

IV. Simulation Results

In this section, we provide MATLAB simulations to demonstrate the effectiveness of the multi-mode defense strategy in different scenarios, as explained below. Some key parameters used in the simulations are: $\rho_a = \rho_d = 0.5\,m$, $C_D = 1.5$, $\bar{v}_a = 6\,m/s$ ($\bar{u}_a = 9\,m/s^2$), $\bar{v}_d = 12.27\,m/s$ ($\bar{u}_d = 18.4\,m/s^2$), $\varrho_d^{int} = 5\,m$, $\rho_p = 45\,m$. The computer specifications used to run these simulations are the same as those used in Section III-E.

We consider a total number of five scenarios (case studies) whose simulation videos are available at (https://youtu.be/cofhjqudT9U). For the interest of space, in this section we provide plots of the simulation of Scenario 3. The description of all scenarios, as well as the detailed results of Scenario 3, are given in the following subsections.

1) Defenders and Attackers are equal in number: We consider three different scenarios.

- Scenario (1): There are 32 attackers that appear, at $t = 0$, to be divided into swarms $\mathcal{A}_{c_1}(0) = \{\mathcal{A}_i | i \in \{1, 2, ..., 20\}\}, \mathcal{A}_{c_2}(0) = \{\mathcal{A}_i | i \in \{21, 22, ..., 29\}\}$ and unclustered attackers $\mathcal{A}_{uc}(0) = \{\mathcal{A}_{30}, \mathcal{A}_{31}, \mathcal{A}_{32}\}$ that are trying to reach the protected area, and 32 defenders that are aiming to prevent the attackers from doing so. In this scenario, after some time, $\mathcal{A}_{c_1}$ splits into 3 smaller swarms and some of the terminal attackers from $\mathcal{A}_{c_2}$ separate into individual risk-taking attackers.
- Scenario (2): There are 20 attackers that are divided into swarms $\mathcal{A}_{c_1}(0) = \{\mathcal{A}_i | i \in \{1, 2, ..., 12\}\}, \mathcal{A}_{c_2}(0) = \{\mathcal{A}_i | i \in \{13, 14, ..., 17\}\}$ and unclustered attackers $\mathcal{A}_{uc}(0) = \{\mathcal{A}_{18}, \mathcal{A}_{19}, \mathcal{A}_{20}\}$. In this scenario, some of the attackers from $\mathcal{A}_{c_1}$ separate as individual risk-taking attackers.
- Scenario (3): At t=0, when the attackers are first identified, they are observed to be distributed as: 2 swarms $A_{c_1}(0) = \{\mathcal{A}_i | i \in \{1, 2, 3, ..., 10\}\}$, $A_{c_2}(0) = \{\mathcal{A}_i | i \in \{11, 12, 13, 14\}\}$, and unclustered attckers $\mathcal{A}_{uc}(0) = \{\mathcal{A}_{15}, \mathcal{A}_{16}\}$.

In the interest of space, we only discuss Scenario 3 in more detail here. For the purpose of demonstration, the motion of the unclustered attackers is simulated under the time-optimal control to reach the protected area. The problem of finding the defenders' assignment to the attackers and the gathering formations is solved using Algorithm 1. This results into two sub-teams of defenders $D_{c_1}(0) = \{\mathcal{D}_{12}, \mathcal{D}_{10}, \mathcal{D}_{16}, \mathcal{D}_{14}, \mathcal{D}_8, \mathcal{D}_7, \mathcal{D}_9, \mathcal{D}_{13}, \mathcal{D}_1, \mathcal{D}_2\}$ and $D_{c_2}(0) = \{\mathcal{D}_{15}, \mathcal{D}_{11}, \mathcal{D}_6, \mathcal{D}_3\}$ being assigned to gather on the time-optimal paths of $\mathcal{A}_{c_1}(0)$ and $\mathcal{A}_{c_2}(0)$, respectively, and 2 individual defenders $\mathcal{D}_4$ and $\mathcal{D}_5$ being assigned to intercept the unclustered attackers $\mathcal{A}_{15}$ and $\mathcal{A}_{16}$, respectively. Figure 10a shows the paths traversed by the players until all defenders' sub-teams gather at their respective desired formations, between the time interval $[0, 77.66]$ sec. As observed, both sub-teams of the defenders are able to successfully gather on the desired formations before respective attackers' swarm could reach there. The paths for the defenders in $\mathcal{D}_{c_1}(0)$ and the attackers in $\mathcal{A}_{c_1}(0)$ during the time interval $[77.66, 130.14]$ sec are shown in Figure 10c. As one can observe, the attackers $\mathcal{A}_{c_1}(0)$ split at $t = t_{se} = 93.12$ sec into two smaller swarms $\mathcal{A}_{c_1}(t_{se}) = \{\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5\}$ and $\mathcal{A}_{c_3}(t_{se}) = \{\mathcal{A}_6, \mathcal{A}_7, \mathcal{A}_8, \mathcal{A}_9\}$, and two outermost attackers, classified as unclustered attackers $\mathcal{A}_{uc}^{(1)}(t_{se}) = \{\mathcal{A}_1, \mathcal{A}_{10}\}$, separate from the rest of the attackers in an attempt to circumvent the oncoming defenders. After solving the rs-MIQCQP (13), the defenders in $\mathcal{D}_{c_1}(0)$ are also divided into two smaller sub-teams $\mathcal{D}_{c_1}(t_{se}) = \{\mathcal{D}_{10}, \mathcal{D}_{16}, \mathcal{D}_{14}, \mathcal{D}_8\}$ and $\mathcal{D}_{c_3}(t_{se}) = \{\mathcal{D}_7, \mathcal{D}_9, \mathcal{D}_{13}, \mathcal{D}_1\}$ and two terminal defenders $\mathcal{D}_{12}$ and $\mathcal{D}_2$. The sub-teams $\mathcal{D}_{c_1}(t_{se})$ and $\mathcal{D}_{c_3}(t_{se})$ are assigned to herd $\mathcal{D}_{c_1}(t_{se})$ and $\mathcal{D}_{c_3}(t_{se})$, respectively. And, the terminal defenders $\mathcal{D}_{12}$ and $\mathcal{D}_2$ are tasked to intercept the unclustered attackers $\mathcal{A}_1$ and $\mathcal{A}_{10}$, respectively. By the time $t = 130.14$ sec the two unclustered attackers are already captured and the two swarms of attackers are also completely enclosed by Closed-StringNets $\mathcal{G}_{sn}^{cl}(\mathcal{D}_{c_1}(t_{se}))$ and $\mathcal{G}_{sn}^{cl}(\mathcal{D}_{c_3}(t_{se}))$. Similarly, as shown in Figure 10b the defenders in $\mathcal{D}_{c_2}(0)$ also successfully enclose the attackers in $\mathcal{A}_{c_2}(0)$ at $t = 146.17$ sec. Finally, as observed in Figure 10d all the enclosed attackers' swarms are herded to the respective closest areas by the Closed-StringNets formed by the defenders' sub-teams. As mentioned also above, simulations for the additional scenarios are provided in the simulation video available at https://youtu.be/cofhjqudT9U.

2) Attackers outnumber the defenders: We also studied the performance of the proposed algorithm in a few scenarios where attackers outnumber the defenders. Particularly, we consider the following two scenarios.

- Scenario (4): There are 16 attackers that are, at $t = 0$, divided into 2 swarms $\mathcal{A}_{c_1}(0) = \{\mathcal{A}_i | i \in \{1, 2, ..., 6\}\}, \mathcal{A}_{c_2}(0) = \{\mathcal{A}_i | i \in \{7, 8, ..., 14\}\}$ and unclustered attackers $\mathcal{A}_{uc}(0) = \{\mathcal{A}_{15}, \mathcal{A}_{16}\}$ and there are only 14 defenders. In this scenario, since the

defenders are short in number by 2 and there are 2 swarms of attackers, resource allocation assigns 5 defenders ($\mathcal{D}_{c_1}(0) = \{\mathcal{D}_5, \mathcal{D}_7, \mathcal{D}_8, \mathcal{D}_{12}, \mathcal{D}_{13}\}$) to $\mathcal{A}_{c_1}$ which has 6 attackers in it and 7 defenders ($\mathcal{D}_{c_2}(0) = \{\mathcal{D}_2, \mathcal{D}_1, \mathcal{D}_9, \mathcal{D}_6, \mathcal{D}_{10}, \mathcal{D}_{11}, \mathcal{D}_{14}\}$) to $\mathcal{A}_{c_2}$ which has 8 attackers in it and the remaining two defenders to intercept the unclustered attackers. As time progresses, at around $t_{se} = 93.58$ sec, $\mathcal{A}_{c_2}(t_{se}^-)$ splits into two smaller swarms $\mathcal{A}_{c_2}(t_{se}) = \{\mathcal{A}_7, \mathcal{A}_8, \mathcal{A}_9, \mathcal{A}_{10}\}$ and $\mathcal{A}_{c_3}(t_{se}) = \{\mathcal{A}_{11}, \mathcal{A}_{12}, \mathcal{A}_{13}, \mathcal{A}_{14}\}$. Again, since $\mathcal{D}_{c_2}(t_{se}^-)$ is short by 1 defender, only 3 defenders ($\mathcal{D}_{c_2}(t_{se}) = \{\mathcal{D}_2, \mathcal{D}_1, \mathcal{D}_9\}$) are assigned to $\mathcal{A}_{c_2}(t_{se}$ and 4 defenders ($\mathcal{D}_{c_3}(t_{se}) = \{\mathcal{D}_6, \mathcal{D}_{10}, \mathcal{D}_{11}, \mathcal{D}_{14}\}$) are assigned to $\mathcal{A}_{c_3}(t_{se})$. The trajectories of the players for this scenario are shown in the simulation video (https://youtu.be/cofhjqudT9U). As one can observer in the video, the defenders are still able to enclose the attackers' swarms successfully and herd them to respective safe areas despite more number of attackers in the attacking swarms. This is because the attackers did not disperse and stayed in compact formations throughout, that the available defenders were capable of enclosing with the given constraints ($\bar{R}$). However, this is a very specific behaviour by the attackers that results in outcomes in favor of the defenders.

- Scenario (5): There are 6 attackers, all of them individual attackers and only 4 defenders. The four attackers ($\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$) approach the protected area from one side and the other two ($\mathcal{A}_5, \mathcal{A}_6$) approach the protected area from the opposite side. Because of the initial states of the defenders, ($\mathcal{D}_2, \mathcal{D}_4, \mathcal{D}_3, \mathcal{D}_1$) are assigned to attackers ($\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$) in that order. After the defender $\mathcal{D}_3$ and $\mathcal{D}_1$ capture their target attackers they get assigned to $\mathcal{A}_6$ and $\mathcal{A}_5$ respectively. Again, the trajectories of the players are shown in the simulation video (https://youtu.be/cofhjqudT9U). As one can observe in the video, despite the reassignment, the attackers $\mathcal{A}_5$ and $\mathcal{A}_6$ are able to reach the protected area. This is because the attackers $A_1 - A_4$ started moving away from the protected area as they saw the defenders coming towards them. By the time the $\mathcal{D}_3$ and $\mathcal{D}_1$ intercepted $\mathcal{A}_3$ and $\mathcal{A}_4$, the defenders had already moved very far from the protected area and hence were not able to come back in time and intercept the remaining two attackers.

These two scenarios show that the success of the defenders when attackers outnumber the defenders is not necessarily govern by the difference in their number but rather by the initial state of the players and how the attackers behave.

V. Conclusions

In this paper, we combine a multi-mode inter-defender collision-aware interception strategy (IDCAIS) with a swarm-herding strategy (StringNet Herding) to provide a multi-mode defense strategy against a wide range of behaviors by the attackers. We provided mixed-integer programs and computationally-efficient heuristics to allocate the interception or herding task to the defenders. Through simulations we showed how the defenders initially attempt to herd the attackers instead of intercepting the risk-averse swarms of the attackers, and how defenders redistribute to sub-teams and reassign either the herding or the interception role to themselves as the attackers split and take on risk-taking or risk-averse roles. The provided heuristics for solving the assignment problems offer a significant reduction in the computational time, by at least a factor of 4-5, while being close to the optimal solution, within 4% error. Future work will focus on considering modeling and measurement uncertainty, as well as extending the formulation to 3D spaces.

References

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] S. H. Lee, "A model predictive control approach to a class of multiplayer minmax differential games," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2016.

[3] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1451–1457, 2017.

[4] A. Pierson, Z. Wang, and M. Schwager, "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 530–537, 2016.

[5] M. Coon and D. Panagou, "Control strategies for multiplayer target-attacker-defender differential games with double integrator dynamics," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 1496–1502.

[6] R. Yan, Z. Shi, and Y. Zhong, "Optimal strategies for the lifeline differential game with limited lifetime," *International Journal of Control*, pp. 1–14, 2019.

[7] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, "Maximum-matching capture strategies for 3d heterogeneous multiplayer reach-avoid games," *arXiv preprint arXiv:1909.11881*, 2019.

[8] D. Shishika, J. Paulos, M. R. Dorothy, M. A. Hsieh, and V. Kumar, "Team composition for perimeter defense with patrollers and defenders," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 7325–7332.

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

[10] D. Shishika, J. Paulos, and V. Kumar, "Cooperative team strategies for multi-player perimeter-defense games," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2738–2745, 2020.

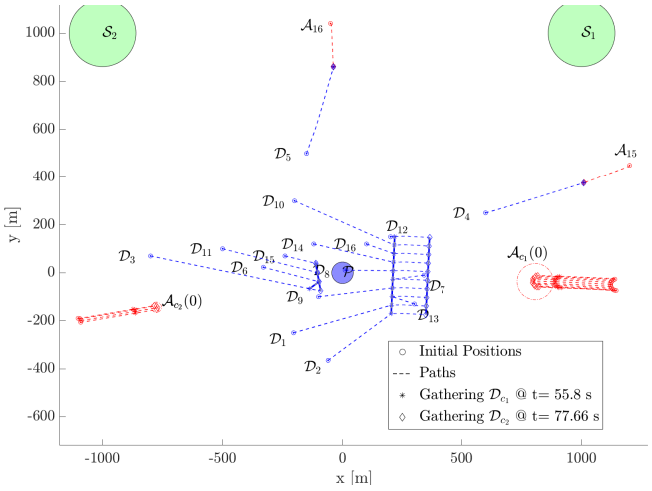
[11] E. Garcia, D. W. Casbeer, and M. Pachter, "Optimal strategies for a class of multi-player reach-avoid differential games in 3d space," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4257–4264, 2020.

[12] V. S. Chipade and D. Panagou, "Idcais: Inter-defender collision-aware interception strategy against multiple attackers," *under review*, 2021. [Online]. Available: http://www-personal.umich.edu/~vishnuc/publications/IDCAIS_under_review.pdf

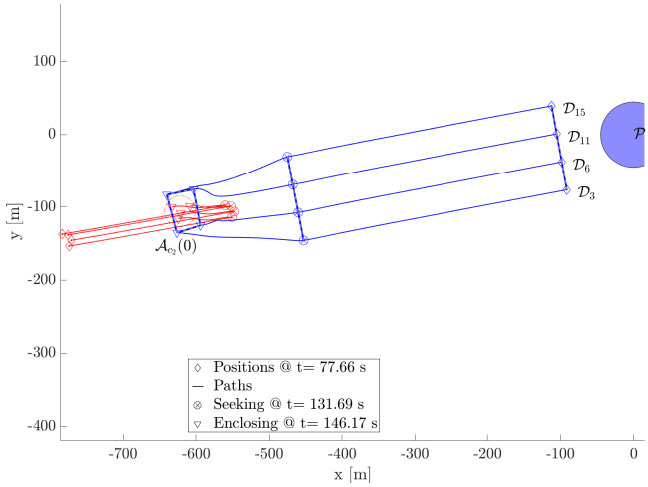
[13] A. A. Paranjape, S.-J. Chung, K. Kim, and D. H. Shim, "Robotic herding of a flock of birds using an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 901–915, 2018.

[14] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, 2018.

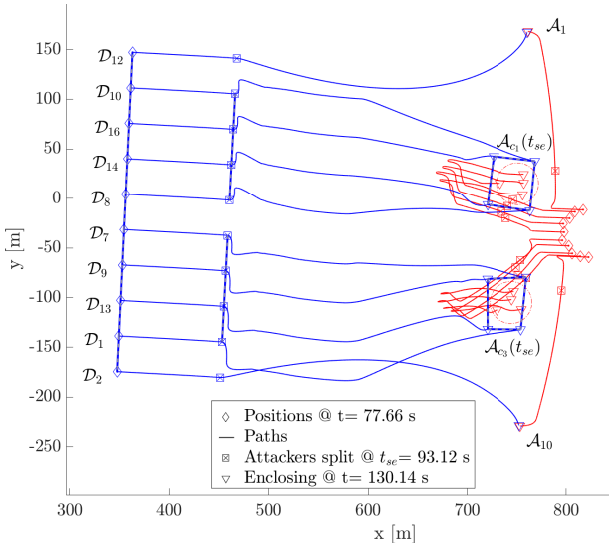
[15] M. A. Haque, A. R. Rahmani, and M. B. Egerstedt, "Biologically inspired confinement of multi-robot systems," *International Journal of Bio-Inspired Computation*, vol. 3, no. 4, pp. 213–224, 2011.



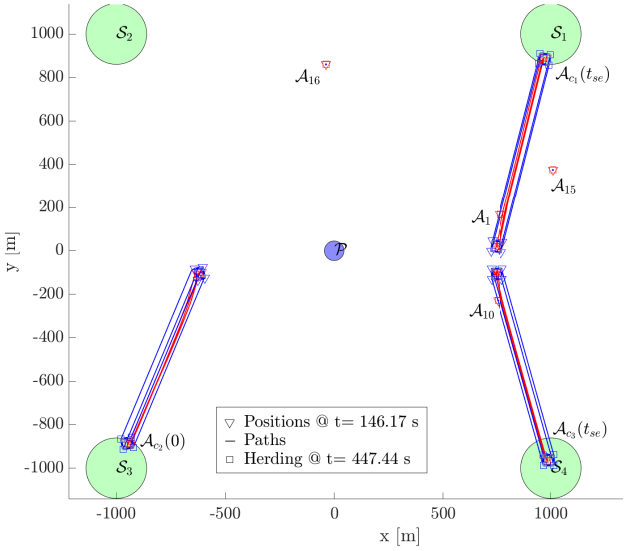
(a) Paths of all agents during time interval $[0, 77.66]$ s



(b) Paths of agents in $\mathcal{D}_{c_2}(0)$ and $\mathcal{A}_{c_2}(0)$ during time interval $[77.66, 146.17]$ s



(c) Paths of agents in $\mathcal{D}_{c_1}(0)$ and $\mathcal{A}_{c_1}(0)$ during time interval $[77.66, 146.17]$ s



(d) Paths of all agents during the herding phase (blue: defenders, red: attackers)

Figure 10: Snapshots of the paths of the agents during multi-mode defense (blue: defenders, red: attackers)

[16] A. Varava, K. Hang, D. Kragic, and F. T. Pokorny, "Herding by caging: a topological approach towards guiding moving agents via mobile robots," in *Proceedings of Robotics: Science and Systems*, 2017.

[17] R. A. Licitra, Z. D. Hutcheson, E. A. Doucette, and W. E. Dixon, "Single agent herding of n-agents: A switched systems approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 374–14 379, 2017.

[18] R. A. Licitra, Z. I. Bell, E. A. Doucette, and W. E. Dixon, "Single agent indirect herding of multiple targets: A switched adaptive control approach," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 127–132, 2018.

[19] P. Deptula, Z. I. Bell, F. M. Zegers, R. A. Licitra, and W. E. Dixon, "Single agent indirect herding via approximate dynamic programming," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7136–7141.

[20] S. Nardi, F. Mazzitelli, and L. Pallottino, "A game theoretic robotic team coordination protocol for intruder herding," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4124–4131, 2018.

[21] V. S. Chipade and D. Panagou, "Herding an adversarial swarm in an obstacle environment," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 3685–3690.

[22] V. S. Chipade and P. Dimitra, "Multi-agent planning and control for swarm herding in 2d obstacle environments under bounded inputs," *Accepted in IEEE Transactions on Robotics*, 2020.

[23] V. S. Chipade and D. Panagou, "Multi-swarm herding: Protecting against adversarial swarms," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5374–5379.

[24] Y. Ho, A. Bryson, and S. Baron, "Differential games and optimal pursuit-evasion strategies," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 385–389, 1965.

[25] H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 4835–4840.

[26] J. S. Jang and C. Tomlin, "Control strategies in multi-player pursuit and evasion game," in *AIAA guidance, navigation, and control conference and exhibit*, 2005, p. 6239.

[27] Y. Wang, L. Dong, and C. Sun, "Cooperative control for multi-player pursuit-evasion games with reinforcement learning," *Neurocomputing*, vol. 412, pp. 101–114, 2020.

[28] Z. Zhou and H. Xu, "Decentralized optimal large scale multi-player pursuit-evasion strategies: A mean field game approach with reinforcement learning," *Neurocomputing*, vol. 484, pp. 46–58, 2022.

[29] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.
[30] R. H. Venkatesan and N. K. Sinha, "The target guarding problem revisited: Some interesting revelations," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1556–1561, 2014.
[31] M. Pachter, E. Garcia, and D. W. Casbeer, "Differential game of guarding a target," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 2991–2998, 2017.
[32] M. W. Harris, "Abnormal and singular solutions in the target guarding problem with dynamics," *Journal of Optimization Theory and Applications*, vol. 184, no. 2, pp. 627–643, 2020.
[33] J. Mohanan, N. Kothuri, and B. Bhikkaji, "The target guarding problem: A real time solution for noise corrupted measurements," *European Journal of Control*, vol. 54, pp. 111–118, 2020.
[34] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1451–1456.
[35] V. S. Chipade, V. A. Marella, and D. Panagou, "Aerial swarm defense by stringnet herding:theory and experiments," *Frontiers in Robotics and AI*, vol. 8, p. 81, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.640446/
[36] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[37] E. Bakolas, "Optimal guidance of the isotropic rocket in the presence of wind," *Journal of Optimization Theory and Applications*, vol. 162, no. 3, pp. 954–974, 2014.
[38] A. Mirjan, A. Federico, D. Raffaello, G. Fabio, and K. Matthias, "Building a bridge with flying robots," in *Robotic Fabrication in Architecture, Art and Design 2016*. Springer, Cham, 2016, pp. 34–47.
[39] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com
[40] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.
[41] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3293–3298.