Multi-Swarm Herding: Protecting against Adversarial Swarms

Vishnu S. Chipade and Dimitra Panagou

***Abstract*— This paper studies a defense approach against one or more swarms of adversarial agents. In our earlier work, we employ a closed formation ('StringNet') of defending agents (defenders) around a swarm of adversarial agents (attackers) to confine their motion within given bounds, and guide them to a safe area. The control design relies on the assumption that the adversarial agents remain close enough to each other, i.e., within a prescribed connectivity region. To handle situations when the attackers no longer stay within such a connectivity region, but rather split into smaller swarms (clusters) to maximize the chance or impact of attack, this paper proposes an approach to learn the attacking subswarms and reassign defenders towards the attackers. We use a 'Density-based Spatial Clustering of Application with Noise (DBSCAN)' algorithm to identify the spatially distributed swarms of the attackers. Then, the defenders are assigned to each identified swarm of attackers by solving a constrained generalized assignment problem. Simulations are provided to demonstrate the effectiveness of the approach.**

I. Introduction

Swarms of low-cost agents such as small aerial robots may pose risk to safety-critical infrastructure such as government facilities, airports, and military bases. Under the assumption of risk-averse and self-interested adversarial agents (attackers) that tend to move away from the defending agents (defenders) and from other dynamic objects, herding can be used as an indirect way of guiding the attackers to a safe area in order to defend a safety-critical area (protected area).

In our recent work [1], [2], we developed a herding algorithm, called 'StringNet Herding', to herd a swarm of attackers away from a protected area. A closed formation ('StringNet') of defending agents connected by string barriers is formed around a swarm of attackers staying together to confine their motion within given bounds, and guide them to a safe area. However, the assumption that the attackers will stay together in a circular region, and they will react to the defenders collectively as a single swarm can be quite conservative in practice.

In this paper, we build upon our earlier work on 'StringNet Herding' [2] and study the problem of defending a protected area from attackers that *may* or *may not* stay together. We propose a 'Multi-Swarm StringNet Herding' approach that uses clustering-based defender assignment, and the 'StringNet Herding' method to herd the adversarial attackers to known safe areas.

1) Related work: Several approaches have been proposed to solve the problem of herding. Some examples are: the n-wavefront algorithm [3], [4], where the motion of the birds on the boundary of the flock is influenced based on the locations of the airport and the safe area; herding via formation control based on a potential-field approach [5]; biologically-inspired "wall" and "encirclement" methods that dolphins use to capture a school of fish [6]; an RRT approach that finds a motion plan for the agents while maintaining a cage of potentials around the sheep [7]; sequential switching among the chased targets [8]. In general, the above approaches suffer from one or more of the following: 1) dependence on knowing the analytical modeling of the attackers' motion, 2) lack of modeling of the adversarial agents' intent to reach or attack a certain protected area, 3) simplified motion and environment models. The proposed 'StringNet Herding' approach relaxes the first and the third issue above, and takes into account the second one for control design.

Clustering of data points is a popular machine learning technique [9]–[13]. Spatial proximity of the agents is crucial for the problem at hand so we focus mostly on the density based approaches such as DBSCAN [11].

Assignment problems have also been studied extensively [14]. In this paper, we are interested in a generalized assignment problem (GAP) [15], in which there are more number of objects than knapsacks to be filled. GAP is known to be NP-hard but there are approximation algorithms to solve an arbitrary instance of GAP [15].

2) Overview of the proposed approach: The proposed approach involves: 1) identification of the clusters (swarms) of the attackers that stay together, 2) distribution and assignment of the defenders to each of the identified swarms of the attackers, 3) use of 'StringNet Herding' approach by the defenders to herd each identified swarm of attackers to the closest safe area.

More specifically, we use the "Density based Spatial Clustering of Application with Noise (DBSCAN)" algorithm [11] to identify the swarms of the attackers in which the attackers stay in a close proximity of the other attackers in the same swarm. We then formulate a generalized assignment problem with additional constraints on the connectivity of the defenders to find which defender should go against which swarm of attackers and herd it to one of the safe areas. This connectivity constrained generalized assignment problem (C2GAP) is modeled as a mixed integer quadratically constrained program

The authors are with 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.

(MIQCP) to obtain an optimal assignment solution. We also provide a hierarchical algorithm to find the assignment quickly, which along with the MIQCP formulation is the major contribution of this paper.

3) Structure of the paper: Section II describes the mathematical modeling and problem statement. The 'StringNet Herding' approach is briefly discussed in Section III. The approach on clustering and the defenders-to-attackers assignment for multiple-swarm herding is discussed in Section IV. Simulations and conclusions are provided in Section V and VI, respectively.

II. Modeling and Problem Statement

Notation: Vectors and matrices are denoted by small and capital bold letters, respectively (e.g., $\mathbf{r}$, $\mathbf{P}$). $\mathbb{Z}_{>0}$ denotes the set of integers greater than 0. $\|.\|$ denotes the Euclidean norm of its argument. $|.|$ denotes the absolute value of a scalar, and cardinality if the argument is a set.

We consider N_a attackers $\mathcal{A}_i$, $i \in I_a = \{1, 2, ..., N_a\}$, and N_d defenders $\mathcal{D}_j$, $j \in I_d = \{1, 2, ..., N_d\}$, operating in a 2D environment $\mathcal{W} \subseteq \mathbb{R}^2$ that contains a protected area $\mathcal{P} \subset \mathcal{W}$, defined as $\mathcal{P} = \{\mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r} - \mathbf{r}_p\| \leq \rho_p\}$, 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 $(\mathbf{r}_p, \rho_p)$ and $(\mathbf{r}_{sm}, \rho_{sm})$ are the centers and radii of the corresponding areas, respectively. 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 $\rho_d \leq \rho_a$, respectively and move under double integrator (DI) dynamics with quadratic drag:

$$\dot{\mathbf{r}}_{ai} = \mathbf{v}_{ai}, \qquad \dot{\mathbf{v}}_{ai} = \mathbf{u}_{ai} - C_D \|\mathbf{v}_{ai}\| \mathbf{v}_{ai}; \tag{1a}$$

$$\dot{\mathbf{r}}_{dj} = \mathbf{v}_{dj}, \qquad \dot{\mathbf{v}}_{dj} = \mathbf{u}_{dj} - C_D \|\mathbf{v}_{dj}\| \mathbf{v}_{dj}; \tag{1b}$$

$$\|\mathbf{u}_{ai}\| \leq \bar{u}_a, \quad \|\mathbf{u}_{dj}\| \leq \bar{u}_d; \tag{1c}$$

where C_D is the drag coefficient, $\mathbf{r}_{ai} = [x_{ai}\ y_{ai}]^T$ and $\mathbf{r}_{dj} = [x_{dj}\ y_{dj}]^T$ are the position vectors of $\mathcal{A}_i$ and $\mathcal{D}_j$, respectively; $\mathbf{v}_{ai} = [v_{x_{ai}}\ v_{y_{ai}}]^T$, $\mathbf{v}_{dj} = [v_{x_{dj}}\ v_{y_{dj}}]^T$ are the velocity vectors, respectively, and $\mathbf{u}_{ai} = [u_{x_{ai}}\ u_{y_{ai}}]^T$, $\mathbf{u}_{dj} = [u_{x_{dj}}\ u_{y_{dj}}]^T$ are the accelerations (the control inputs), respectively. This model poses a speed bound on each player with limited acceleration control, i.e., $v_{ai} = \|\mathbf{v}_{ai}\| < \bar{v}_a = \sqrt{\frac{\bar{u}_a}{C_d}}$ and $v_{dj} = \|\mathbf{v}_{dj}\| < \bar{v}_d = \sqrt{\frac{\bar{u}_d}{C_d}}$. The defenders are assumed to be faster than the attackers, i.e., $\bar{v}_a < \bar{v}_d$ (i.e., $\bar{u}_a < \bar{u}_d$).

***Assumption** 1:* 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 $\mathcal{Z}_d = \{\mathbf{r} \in \mathbb{R}^2 | \ \|\mathbf{r} - \mathbf{r}_{pa}\| \leq \varrho_d\}$ 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 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 the attacker $\mathcal{A}_i$'s sensing zone.

The attackers aim to reach the protected area $\mathcal{P}$. The attackers may use flocking controllers [16] to stay together, or they may choose to split into different smaller swarms [17], [18]. The defenders aim to herd each of these attackers to one of the safe areas in $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{N_s}\}$ before they reach $\mathcal{P}$. We consider the following problems.

***Problem** 1 (Swarm Identification):* Identify the swarms $\{\mathcal{A}_{c_1}, \mathcal{A}_{c_2}, ..., \mathcal{A}_{c_{N_{ac}}}\}$ of the attackers for some unknown $N_{ac} \geq 1$ such that attackers in the same swarm $\mathcal{A}_{c_k}$, and only them, satisfy prescribed conditions on spatial proximity, where $\mathcal{A}_{c_k} = \{\mathcal{A}_i | i \in A_{c_k}\}$, $A_{c_k} \subseteq I_a$, for all $k \in I_{ac} = \{1, 2, ..., N_{ac}\}$.

***Problem** 2 (Multi-Swarm Herding):* Find subgroups $\{\mathcal{D}_{c_1}, \mathcal{D}_{c_2}, ..., \mathcal{D}_{c_{N_{ac}}}\}$ of the defenders and their assignment to the attackers' swarms, such that the defenders in the same subgroup are connected via string barriers to enclose and herd the assigned attackers' swarm.

III. Herding a Single Swarm of Attackers

To herd a swarm of attackers to $\mathcal{S}$, we use 'StringNet Herding', developed in [2]. StringNet is a closed net of strings formed by the defenders as shown in Fig. 1. The strings are realized as impenetrable and extendable line barriers (e.g., spring-loaded pulley and a rope or other similar mechanism [19]) 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 $\bar{R}_{sb}>0$. 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. The underlying graph structure for the two different "StringNet" formations defined for a subset of defenders $\mathcal{D}' = \{\mathcal{D}_j \mid j \in I'_d\}$, where $I'_d \subseteq I_d$, are defined as follows:

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

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

The StringNet herding consists of four phases: 1) gathering, 2) seeking, 3) enclosing, and 4) herding to a safe area. These phases are discussed as follows.

1) Gathering: We assume that the attackers start as single swarm that stays together and they may start splitting into smaller groups as they sense the defenders in their path. The aim of the defenders is to converge to an open formation $\mathscr{F}^g_d$ centered at the gathering center $\mathbf{r}_{df^g}$ located on the expected path of the attackers, where the expected path is defined as the shortest path of the attackers to the protected area, before the attackers reach $\mathbf{r}_{df^g}$. Let $\mathscr{R}_d(N_a) : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ be the resource allocation function that outputs the number of the defenders that can be assigned to the given N_a attackers. The open formation $\mathscr{F}^g_d$ is characterized by the positions $\boldsymbol{\xi}^g_l$, for all $l \in I_{dc_0} = \{1, 2, ..., \mathscr{R}_d(N_a)\}$, and is chosen to be a straight

line formation[1] (see Fig. 1). Once the defenders arrive at these positions, the defenders get connected by strings as follows: the defender at $\boldsymbol{\xi}_l^g$ gets connected to the defender at $\boldsymbol{\xi}_{l+1}^g$ for all $l \in I_{dc_0}^- = \{1, 2, ..., \mathscr{R}_d(N_a)-1\}$ (see Fig. 1). The angle made by the normal to the line joining $\boldsymbol{\xi}_1^g$ and $\boldsymbol{\xi}_{N_d}^g$ (clockwise from $\boldsymbol{\xi}_1^g$, see Fig. 1) is the orientation ϕ of the formation. The formation $\mathscr{F}_d^g$ is chosen such that its orientation is toward the attackers on their expected path (defined above), see the blue formation in Fig 1. The desired positions $\boldsymbol{\xi}_l^g$ on $\mathscr{F}_d^g$ centered at $\mathbf{r}_{df^g}$ are: $\boldsymbol{\xi}_l^g = \mathbf{r}_{df^g} + R_l\hat{\mathbf{o}}(\theta_{df^g} + \frac{\pi}{2})$, for all $l \in I_{dc_0}$; where $R_l = 0.5\,(\mathscr{R}_d(|A_{c_k}|) - 2l + 1)\,\bar{R}_{sb}$, $\hat{\mathbf{o}}(\theta) = [\cos(\theta),\ \sin(\theta)]^T$ is the unit vector making an angle θ with x-axis, $\theta_{df^g} = \theta^*_{a_{cm}} + \pi$, where $\theta^*_{a_{cm}}$ is the angle made by the line segment joining the attackers' center of mass (ACoM) to the center of the protected area (the shortest path from the initial position of ACoM to $\mathcal{P}$) with x-axis. These positions are static, i.e., $\dot{\boldsymbol{\xi}}_l^g = \ddot{\boldsymbol{\xi}}_l^g = \mathbf{0}$. The gathering center $\mathbf{r}_{df^g} = \rho_{df}^g\hat{\mathbf{o}}(\theta_{df^g})$ is such that $\rho_{df}^g > \rho_p$.

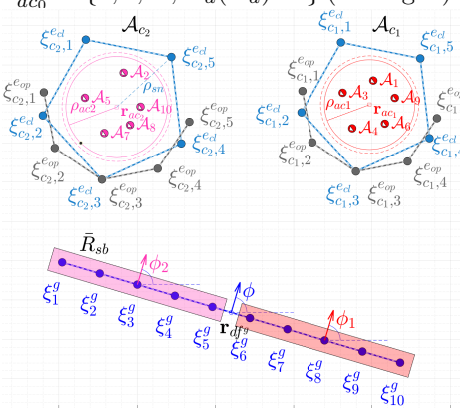

Fig. 1: Defenders' assignment

As discussed in Algorithm 1 in [2], we design a time-optimal motion plan so that the defenders gather at the desired formation $\mathscr{F}_d^g$ as early as possible and before the attackers reach close to $\mathscr{F}_d^g$. The idea in Algorithm 1 [2] is to iteratively solve a mixed integer quadratic program (MIQP) until a gathering center for the gathering formation is found which is as far as possible from the protected area and such that the defenders are able to gather at the formation $\mathscr{F}_d^g$ centered at the gathering center with bounded acceleration before the attackers can.

2) Seeking: After the defenders accomplish gathering, suppose a group of defenders $\mathcal{D}_{c_k} = \{\mathcal{D}_j | j \in D_{c_k}\}$, $D_{c_k} \subseteq I_d$, is tasked to herd a swarm of attackers $\mathcal{A}_{c_k} = \{\mathcal{A}_i | i \in A_{c_k}\}$, $A_{c_k} \subseteq I_a$, the details are discussed later in Section IV. Denote $I_{dc_k} = \{1, 2, ..., |\mathcal{D}_{c_k}|\}$ and let $\beta_k : I_{dc_k} \to D_{c_k}$ be the mapping that gives the indexing order of the defenders in $\mathcal{D}_{c_k}$ on the Open-StringNet line formation $\mathscr{F}_{dc_k}^s$ (similar to $\mathscr{F}_d^g$). In the seeking phase, the defenders in $\mathcal{D}_{c_k}$ maintain the line formation $\mathscr{F}_{dc_k}^s$ and try to get closer to the swarm of attackers $\mathcal{A}_{c_k}$ by using state-feedback, finite-time convergent, bounded control laws as discussed in [2]. The control actions in [2] for the defenders in $\mathcal{D}_{c_k}$ are modified to incorporate collision avoidance from the other StringNet formations formed by $\mathcal{D}_{c_{k'}}$, for $k' \neq k$.

3) Enclosing (Closed-StringNet formation): Once the Open-StringNet formation reaches close to the attackers' formation, the defenders start enclosing the attackers by moving to their desired positions on the enclosing formations while staying connected to their neighbors. We choose two formations for this phase that the defenders sequentially achieve: 1) Semi-circular Open-StringNet formation ($\mathscr{F}_{dc_k}^{e_{op}}$), 2) Circular Closed-StringNet formation ($\mathscr{F}_{dc_k}^{e_{cl}}$). The intermediate Semi-circular Open-StringNet formation is chosen to allow smooth convergence of the defenders to their desired positions on the Circular Closed-StringNet $\mathscr{F}_{dc_k}^{e_{cl}}$ while keeping the abrupt distortions to the formation small.

The desired position $\boldsymbol{\xi}_{c_k,l}^{e_{op}}$ on the Open-StringNet formation $\mathscr{F}_{dc_k}^{e_{op}}$ (Fig. 1) is chosen on the circle with radius ρ_{sn_k} centered at $\mathbf{r}_{ac_k}$ as: $\boldsymbol{\xi}_{c_k,l}^{e_{op}} = \mathbf{r}_{ac_k} + \rho_{sn_k}\hat{\mathbf{o}}(\theta^{e*}_{df_k} + \frac{\pi}{2} + \frac{\pi(l-1)}{|\mathcal{D}_{c_k}|-1})$ for all $l \in I_{dc_k}$, where $\theta^{e*}_{df_k}$ is equal to the orientation ϕ_k of the defenders' group $\mathcal{D}_{c_k}$ at the beginning of the enclosing phase, $\mathbf{r}_{ac_k} = \sum_{i\in I_{ac_k}} \frac{\mathbf{r}_{ai}}{|\mathcal{A}_{c_k}|}$ is the center of mass of $\mathcal{A}_{c_k}$. The radius ρ_{sn_k} should satisfy, $\bar{\rho}_{ac_k} + b_d < \rho_{sn_k}$, where $\bar{\rho}_{ac_k}$ is maximum radius of swarm $\mathcal{A}_{c_k}$. The parameter b_d is the tracking error for the defenders in this phase [2]. Similarly, the desired positions $\boldsymbol{\xi}_{c_k,l}^{e_{cl}}$ on the Closed-StringNet formation $\mathscr{F}_{dc_k}^{e_{cl}}$ are $\boldsymbol{\xi}_{c_k,l}^{e_{cl}} = \mathbf{r}_{ac_k} + \rho_{sn_k}\hat{\mathbf{o}}(\theta^{e*}_{df_k} + \frac{\pi(2l-1)}{|\mathcal{D}_{c_k}|})$, for all $l \in I_{dc_k}$. Both formations move with the same velocity as that of the attackers' center of mass, i.e., $\dot{\boldsymbol{\xi}}_{c_k,l}^{e_{op}} = \dot{\boldsymbol{\xi}}_{c_k,l}^{e_{cl}} = \dot{\mathbf{r}}_{ac_k}$.

The defenders $\mathcal{D}_{c_k}$ first track the desired goal positions $\boldsymbol{\xi}_{c_k,l}^{e_{op}}$ by using the finite-time convergent, bounded control actions given in [2]. Once the defender $\mathcal{D}_{\beta_k(1)}$ and $\mathcal{D}_{\beta_k(|\mathcal{D}_{c_k}|)}$ reach within a distance of b_d from $\boldsymbol{\xi}_{c_k,1}^{e_{op}}$ and $\boldsymbol{\xi}_{c_k,|\mathcal{D}_{c_k}|}^{e_{op}}$, i.e., $\left\|\mathbf{r}_{d\beta_k(1)} - \boldsymbol{\xi}_{c_k,1}^{e_{op}}\right\| < b_d$ and $\left\|\mathbf{r}_{d\beta_k(|\mathcal{D}_{c_k}|)} - \boldsymbol{\xi}_{c_k,|\mathcal{D}_{c_k}|}^{e_{op}}\right\| < b_d$, respectively, the desired goal positions are changed from $\boldsymbol{\xi}_{c_k,l}^{e_{op}}$ to $\boldsymbol{\xi}_{c_k,l}^{e_{cl}}$ for all $l \in I_{dc_k}$. The StringNet is achieved when $\left\|\mathbf{r}_{d\beta_k(l)} - \boldsymbol{\xi}_{c_k,l}^{e_{cl}}\right\| \leq b_d$ for all $l \in I_{dc_k}$ during this phase.

4) Herding (moving the Closed-StringNet to safe area): Once a group $\mathcal{D}_{c_k}$ forms a StringNet around a swarm of attackers $\mathcal{A}_{c_k}$, they move while tracking a desired rigid closed circular formation $\mathscr{F}_{dc_k}^h$ centered at a virtual agent $\mathbf{r}_{df_k^h}$ as discussed in [2]. The swarm is herded to the closest safe area $S_{\varsigma(k)}$, where $\varsigma(k) = \arg\min_{m\in I_s} \left\|\mathbf{r}_{df_k^h} - \mathbf{r}_{sm}\right\|$.

IV. Multi-Swarm Herding

We consider that the attackers split into smaller groups as they sense the defenders in their path, to maximize the chance of at least some attackers reaching the protected area by circumnavigating the oncoming defenders. To respond to such strategic movements of the attackers, the defenders need to collaborate intelligently. In the approach presented in this paper, the defenders first identify the spatial clusters of the attackers. Then, the

[1]This is a better choice compared to a semicircular formation [2]. 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.

defenders distribute themselves into smaller connected groups and these connected groups are assigned to the herd different spatial clusters (swarms) of the attackers to safe areas. In the connected group of defenders, the defenders are already connected via string barriers and have already established an Open-StringNet formation. In the next subsections, we discuss the clustering and the defender to swarm assignment algorithms.

A. Identifying Swarms of the Attackers

To identify the spatially distributed clusters (swarms) of the attackers within a reasonable computational time, the defenders use the Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [11]. Given a set of points, DBSCAN algorithm finds clusters of high density points (points with many nearby neighbors), and marks the points as outliers if they lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN algorithm can identify clusters of any shape in the data and requires two parameters that define the density of the points in the clusters: 1) ε_{nb} (radius of the neighborhood of a point), 2) m_{pts} (minimum number of points in ε_{nb}-neighborhood of a point).

Let $d(\mathbf{x}_{ai}, \mathbf{x}_{ai'}) = \sqrt{(\mathbf{x}_{ai} - \mathbf{x}_{ai'})^T \mathbf{M} (\mathbf{x}_{ai} - \mathbf{x}_{ai'})}$ be the weighted distance between two attackers, where $\mathbf{x}_{ai} = [\mathbf{r}_{ai}^T, \mathbf{v}_{ai}^T]^T$ and $\mathbf{M}$ is a weighing matrix defined as $\mathbf{M} = diag([1, 1, \varphi, \varphi])$, where φ weights relative velocity against relative position. We choose $\varphi < 1$ because relative position is more important in a spatial cluster than the velocity alignment at a given time instance. The ε_{nb}-neighborhood of an attacker $\mathcal{A}_i$ is then defined as the set of points $\mathbf{x} \in \mathbb{R}^4$ such that $d(\mathbf{x}_{ai}, \mathbf{x}) < \varepsilon_{nb}$.

The largest circle inscribed in the largest Closed-StrignNet formation formed by the N_d defenders has radius $\bar{\rho}_{ac} = \frac{\bar{R}_{sb}}{2} \cot(\frac{\pi}{N_d})$. Maximum radius of any cluster with N_a points identified by DBSCAN algorithm with parameters ε_{nb} and m_{pts} is $\frac{\varepsilon_{nb}(N_a - 1)}{m_{pts} - 1}$. If all of the attackers were to be a single swarm enclosed inside the region with radius $\bar{\rho}_{ac}$ then we would require ε_{nb} to be greater than $\frac{\bar{\rho}_{ac}(m_{pts}-1)}{N_a - 1}$ in order identify them as a single cluster. So we choose $\varepsilon_{nb} = \frac{\bar{\rho}_{ac}(m_{pts}-1)}{N_a - 1}$ and since we want to identify even clusters with as low as 3 agents we need to choose $m_{pts} = 3$. For these parameters, we have:

***Lemma** 1:* Let $\{\mathcal{A}_{c_1}, \mathcal{A}_{c_2}, ..., \mathcal{A}_{c_{N_{ac}}}\}$ be the clusters identified by DBSCAN algorithm with $\varepsilon_{nb} = \frac{\bar{\rho}_{ac}}{N_a - 1} \lfloor \frac{m_{pts}}{2} \rfloor$. For all $k \in I_{ac} = \{1, 2, ..., N_{ac}\}$, the radius of the cluster $\mathcal{A}_{c_k}$, ρ_{ac_k}, satisfies $\rho_{ac_k} = \max_{i \in I_{ac_k}} \|\mathbf{r}_{ai} - \mathbf{r}_{ac_k}\| \leq \frac{\bar{R}_{sb}}{2} \cot\left(\frac{\pi}{|\mathcal{A}_{c_k}|}\right)$, if $|\mathcal{A}_{c_k}| > 3$ and $N_a = N_d$.

As the number of attackers increases, the computational cost for DBSCAN becomes higher and looses its practical usefulness. Furthermore, the knowledge of the clusters is only required by the defenders when a swarm of attackers does not satisfy the assumed constraint on its connectivity radius. So the DBSCAN algorithm is run only for swarms of attackers $\mathcal{A}_{c_k}$ for some $k \in I_{ac}$ whenever the connectivity constraint is violated by them i.e., when the radius of the swarm of attackers $\mathcal{A}_{c_k}$ defined as $\rho_{ac_k} = \max_{i \in I_{ac_k}} \|\mathbf{r}_{ai} - \mathbf{r}_{ac_k}\|$ exceeds the value $\bar{\rho}_{ac_k} = \frac{\bar{R}_{sb}}{2} \cot\left(\frac{\pi}{N_d}\right) \frac{|\mathcal{A}_{c_k}| - 1}{N_a - 1}$.

B. Defender Assignment to the Swarms of Attackers

As the initial swarm of attackers splits into smaller swarms, the defenders must distribute themselves into smaller groups and assign the attackers' swarms (clusters) to these groups in order to enclose these swarms and subsequently herd them to the closest safe area. Let $\mathcal{A}_c = \{\mathcal{A}_{c_1}, \mathcal{A}_{c_2}, \ldots, \mathcal{A}_{c_{N_{ac}}}\}$ be a set of swarms of the attackers after a split event has happened at time t_{se}. We assume that none of the swarms in $\mathcal{A}_c$ is a singular one, i.e., $|\mathcal{A}_{c_k}| > 2$ for all $k \in I_{ac}$. We formally define the defender to attackers' swarm assignment as:

***Definition** 3 (Defender-Swarm Assignment):* A set $\beta = \{\beta_1, \beta_2, ...\beta_{N_{ac}}\}$ of mappings $\beta_k : \{1, 2, ..., \mathscr{R}_d(|\mathcal{A}_{c_k}|)\} \rightarrow I_d$, where β_k gives the indices of the defenders assigned to the swarm $\mathcal{A}_{c_k}$ for all $k \in I_{ac}$.

We want to find an assignment that minimizes the sum of distances of the defenders from the centers of the attackers' swarms to which they are assigned. This ensures that the collective effort needed by all the defenders is minimized when enclosing the swarms of the attackers. For successful enclosing of the newly formed attacking swarms, it is required that all the defenders that are assigned to a particular swarm of the attackers are neighbors of each other, are already connected to each other via string barriers and the underlying graph is an Open-StringNet. 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 $\mathcal{A}_{c_k}$. This is to ensure that there are adequate number of defenders to go after each attacker in the event the attackers in swarm $\mathcal{A}_{c_k}$ disintegrate into singular swarms. In the case of singular swarms, herding may not be the most economical way of defense. The case of singular swarms will be studied in the future work.

This assignment problem is closely related to generalized assignment problem (GAP) [15], in which n objects are to be filled in m knapsacks ($n \geq m$). This problem is modeled as a GAP with additional constraints on the objects (defenders) that are assigned to a given knapsack (attackers' swarm). We call this constrained assignment problem as connectivity constrained generalized assignment problem (C2GAP) and provide a mixed integer quadratically constrained program (MIQCP) to find the optimal assignment as:

$$\text{Minimize } J = \sum_{k=1}^{N_{ac}} \sum_{j=1}^{N_d} \|\mathbf{r}_{ac_k} - \mathbf{r}_{dj}\| \, \delta_{jk} \tag{2a}$$

$$\text{Subject to } \sum_{k \in I_{ac}} \delta_{jk} = 1, \quad \forall j \in I_d; \tag{2b}$$

$$\sum_{j \in I_d} \delta_{jk} = \mathscr{R}_d(|\mathcal{A}_{c_k}|), \quad \forall k \in I_{ac}; \tag{2c}$$

$$\sum_{j \in I_{dc_0}^-} \delta_{jk} \delta_{(j+1)k} \geq \mathscr{R}_d(|\mathcal{A}_{c_k}|) - 1, \quad \forall k \in I_{ac}; \tag{2d}$$

$$\sum_{k \in I_{ac}} \sum_{j \in I_d} \delta_{jk} = \mathscr{R}_d(N_a); \tag{2e}$$

$$\delta_{jk} \in \{0, 1\}, \quad \forall j \in I_d, k \in I_{ac}; \tag{2f}$$

where δ_{jk} is a decision variable which is equal to 1 when the defender $\mathcal{D}_j$ is assigned to the swarm $\mathcal{A}_{c_k}$ and 0 otherwise. The constraints (2b) ensure that each defender is assigned to exactly one swarm of the attackers, the capacity constraints (2c) ensure that for all $k \in I_{ac}$ swarm $\mathcal{A}_{c_k}$ has exactly $\mathscr{R}_d(|\mathcal{A}_{c_k}|)$ defenders assigned to it, the quadratic constraints (2d) ensure that all the defenders assigned to swarm $\mathcal{A}_{c_k}$ are connected together with an underlying Open-StringNet for all $k \in I_{ac}$ and the constraint (2e) ensures that all the $\mathscr{R}_d(N_a)$ defenders are assigned to the attackers' swarms. This MIQCP can be solved using a MIP solver Gurobi [20]. As shown in an instance of the defender-swarm assignment in Fig. 1, the defenders at $\boldsymbol{\xi}_l^g$ for $l \in \{1, 2, ..., 5\}$ are assigned to swarm $\mathcal{A}_{c_2}$ and those at $\boldsymbol{\xi}_l^g$ for $l \in \{6, 7, ..., 10\}$ are assigned to swarm $\mathcal{A}_{c_1}$.

C. Heuristic to find Defender-Swarm Assignment

Finding the optimal defender-swarm assignment by solving the MIQCP (2) may not be real-time implementable for a large number of agents (> 100). So, we develop a computationally efficient heuristic called hierarchical approach to find defender-swarm assignment. A large dimensional assignment problem is split into smaller, low-dimensional assignment problems that can be solved optimally and quickly. Specifically we split the attackers' clusters into smaller groups such that each group of clusters has smaller than or equal to $\underline{N}_{ac} (\leq N_{ac})$ clusters. Similarly defenders are divided into corresponding smaller groups and multiple smaller MIQCPs are solved to assign defenders from smaller groups to the corresponding group of attackers' clusters. Due to limited space, the specific details of how these groups are formed are provided in the detailed arXiv version of this paper [21]. As shown in Figure 2, the average computation time over a number of cluster configurations and initial conditions for the hierarchical approach (heuristic) to assignment is significantly smaller than that of the MIQCP formulation and also the cost of the hierarchical algorithm is very close to the optimal cost (MIQCP).

V. Simulations

We provide a simulation of 18 defenders herding 18 attackers to $\mathcal{S}$ with bounded control inputs. Figure 3 shows the snapshots of the paths taken by all agents. The positions and paths of the defenders are shown in blue color, and that of the attackers in red. The string-barriers between the defenders are shown as wide solid blue lines with white dashes in them.

Snapshot 1 shows the paths during the gathering phase. As observed the defenders are able to gather at a location on the shortest path of the attackers to the protected area before the attacker reach there. Five attackers are already separated from the rest thirteen in reaction to the incoming defenders in their path. The defenders have identified two swarms of the attackers $\mathcal{A}_{c_1}$ and $\mathcal{A}_{c_2}$ at the end of the gathering phase and assign two

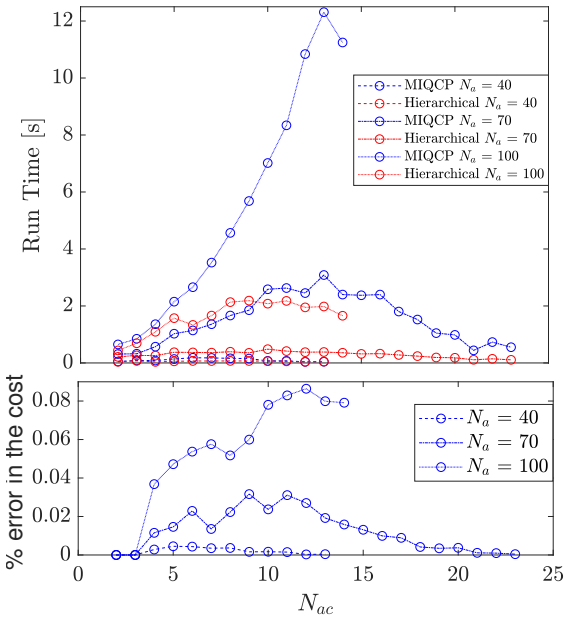

Fig. 2: Comparison of the MIQCP and the Heuristic

subgroups $\mathcal{D}_{c_1}$ and $\mathcal{D}_{c_2}$ of the defenders to $\mathcal{A}_{c_1}$ and $\mathcal{A}_{c_2}$. As shown in snapshot 2, $\mathcal{D}_{c_1}$ and $\mathcal{D}_{c_2}$ seek $\mathcal{A}_{c_1}$ and $\mathcal{A}_{c_2}$, but the attackers in swarm $\mathcal{A}_{c_2}$ further start splitting and the defenders identify this newly formed $\mathcal{A}_{c_2}$ and $\mathcal{A}_{c_3}$ at time $t = 120.11s$. The group $\mathcal{D}_{c_2}$ is then split into two subgroups $\mathcal{D}_{c_2}$ and $\mathcal{D}_{c_3}$ of appropriate sizes and assigned to the new swarms $\mathcal{A}_{c_2}$ and $\mathcal{A}_{c_3}$ after solving (2).

Snapshot 3 shows how the 3 subgroups of the defenders are able to enclose the the identified 3 swarms of the attackers by forming Closed-StringNets around them. Snapshot 4 shows how all the three enclosed swarms of the attackers are taken to the respective closest safe areas while each defenders' group ensures collision avoidance from other defenders' groups. Additional simulations can be found at https://tinyurl.com/yypb2yv9.

VI. Conclusions

We proposed a MIQCP to solve a clustering-based, connectivity-constrained assignment problem that distributes and assigns groups of defenders against swarms of the attackers, to herd them to the closest safe area using 'StringNet Herding' approach. We also provide a heuristic for the defender-swarm assignment, based on the optimal MIQCP, that finds the assignment quickly. Simulations show how this proposed method improves the original 'StringNet Herding' method and enables the defenders to herd all the attackers to safe areas even though the attackers start splitting into smaller swarms in reaction to the defenders.

References

[1] 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.

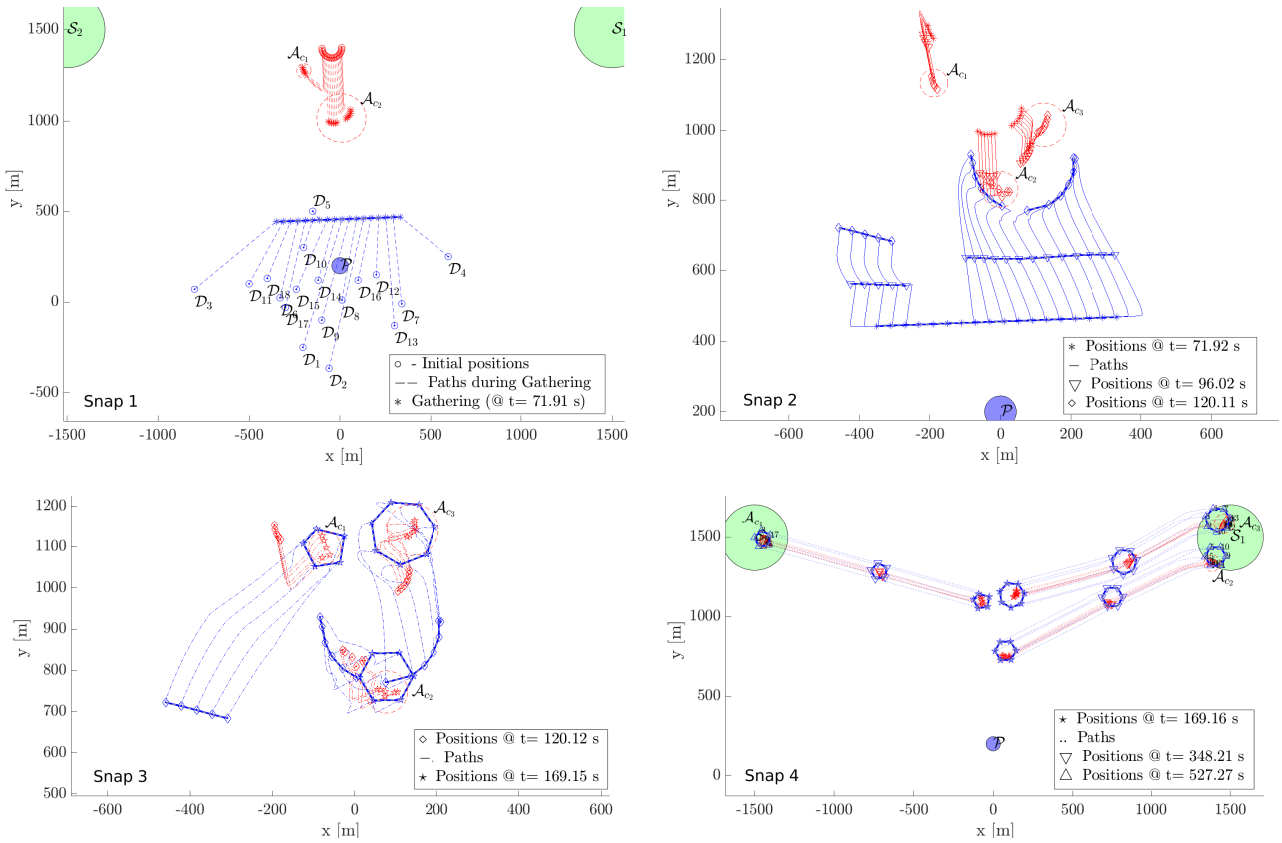

Fig. 3: Snapshots of the paths of the agents during Multi-Swarm StringNet Herding

[2] ——, "Multi-agent planning and control for swarm herding in 2d obstacle environments under bounded inputs," *(under review)*, 2020. [Online]. Available: http://www-personal.umich.edu/~dpanagou/assets/documents/VChipade_under_review.pdf

[3] S. Gade, A. A. Paranjape, and S.-J. Chung, "Herding a flock of birds approaching an airport using an unmanned aerial vehicle," in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 1540.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[10] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[12] L. O'callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering," in *Proceedings 18th International Conference on Data Engineering*. IEEE, 2002, pp. 685–694.

[13] R. Sharan and R. Shamir, "Click: a clustering algorithm with applications to gene expression analysis," in *Proc Int Conf Intell Syst Mol Biol*, vol. 8, no. 307, 2000, p. 16.

[14] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment problems, revised reprint*. Siam, 2012, vol. 106.

[15] T. Öncan, "A survey of the generalized assignment problem and its applications," *INFOR: Information Systems and Operational Research*, vol. 45, no. 3, pp. 123–141, 2007.

[16] B. Dai and W. Li, "Flocking of multi-agents with arbitrary shape obstacle," in *Proceedings of the 33rd Chinese Control Conference*. IEEE, 2014, pp. 1311–1316.

[17] R. Goel, J. Lewis, M. Goodrich, and P. Sujit, "Leader and predator based swarm steering for multiple tasks," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 3791–3798.

[18] K. Raghuwaiya, J. Vanualailai, and B. Sharma, "Formation splitting and merging," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 461–469.

[19] 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.

[20] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com

[21] V. S. Chipade and D. Panagou, "Multi-swarm herding: Protecting against adversarial swarms," *Accepted in CDC 2020, arXiv preprint arXiv:2007.04407*, 2020.