Automatic Identification of the Leader in a Swarm using an Optimized Clustering and Probabilistic Approach

Ajitesh Singh and Panagiotis Artemiadis*, *IEEE Senior Member*

***Abstract*— Collective behavior appearing abundantly in nature at various scales, such as swarming of insects, flocking of birds or schooling of fish, has motivated an interdisciplinary research thrust towards artificial swarms thanks to its numerous advantages. Although a lot of work has been done on coming up with controllers for these multi-agent systems towards optimizing coordination and planning, the inverse problem of identifying behaviors and models from observing a swarm is still not explored enough. Efficient tools for the analysis of the complex spatial-temporal dataset an observed swarm generates, are still missing. In this paper, we propose a methodology to solve the problem of identifying the leader in a swarm governed by leader-follower control framework using only the macroscopic view of the entire swarm. A methodology that combines clustering and probability methods is used to analyze the spatial-temporal data of a swarm of robots to narrow down on a subset of the robots that includes the leader. The clustering parameters are optimized over multiple simulated behaviors. The results show that this automated system can narrow down on a subset of agents (cluster) that includes the leader of the swarm with high accuracy. Applications of the proposed methodology include automatic identification of leader and swarm dynamics in both artificial and biological swarms, which can lead to a better understanding of collective behaviors and predictions of future behaviors.**

I. INTRODUCTION

Collective behavior appearing abundantly in nature at various scales, such as swarming of insects, flocking of birds or schooling of fish, has motivated an interdisciplinary research thrust towards artificial swarms thanks to its numerous advantages. Recent advances in computing, sensing, actuation and control technologies are currently enabling the development of swarms of aerial and ground vehicles, varying in complexity, size and overall scale. For the development of artificial swarms, many researchers have taken inspiration from biological species in nature for the design of collective behaviors of multi-agent systems. Some of the inspirations include flocking of birds [1], swarming of bees [2], and ant colonies [3]. Many parallels can be drawn between biological systems and multi-agent systems, such as that both consist of a large number of biological organisms and robotic agents, while, each individual determines their motion based on local sensing information and communication.

Since swarm robotics involve multiple robots, it makes them a good candidate in a wide range of application like reconnaissance, environmental monitoring, tracking, exploration, search and pursuit-evasion, infrastructure support, protection, and even space exploration [4]. Over the years, the field of swarm robotics has risen from a theoretical possibility to an actual system implemented on real robots in real-world settings.

The control of the multi-agent system however poses its own challenges, since it usually involves multiple objectives, including avoiding collision, maintaining cohesion, and alignment among the agents. There has been a lot of development in designing algorithms for cooperative control of a group of robots including behavior-based controllers [5, 6], virtual structural approaches [7, 8], leader-follower networks [9], potential field approaches [10, 11] and consensus-based approaches [12]–[14].

The local interaction of robots with each other as well as with the environment is one of the basic features of a multi-agent system. In such systems, robots usually make decisions independently governed by local interaction rules for further actions. However, the majority of applications of swarm robotics require a coordinated movement of the swarm. To solve this problem researchers have used a leader-follower framework where, a robot, i.e., the leader, moves along a predefined trajectory that dictates the overall swarm's behavior. The leader is also not affected by the surrounding agents, but it still affects them. Even though over-reliance on a single agent for achieving a common goal in leader–follower frameworks may be undesirable, this approach is appreciated for its simplicity and scalability. For these reasons, having a leader provides an advantage in coordinating a swarm.

As a swarm evolves, multiple clusters are formed due to the local interaction among the agents as well with the leader. Along with that, a spatial-temporal dataset including the states of the agents in the swarm is also created. To analyze this dataset, clustering techniques have been proposed in the past. A method for recognizing moving clusters (two consecutive spatial clusters sharing a large number of common objects) has been proposed in [15]. Another method where the future time splitting and merging of a cluster was used to recognize moving clusters in the spatial–temporal dataset was proposed in [16]. Birant et. al. presented a modification of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [17], the new density-based clustering algorithm ST-DBSCAN, which clusters spatial–temporal dataset based on its non-spatial, spatial, and temporal features [18]. However, none of the above methods were used in spatial-temporal datasets of swarm that have a leader, a characteristic that is usually found

*This work is supported by the U.S. Air Force Office of Scientific Research (AFOSR) award FA9550-18-1-0221.

Ajitesh Singh and Panagiotis Artemiadis are with the Mechanical Engineering Department, at the University of Delaware, Newark, DE 19716, USA. ajiteshs@udel.edu, partem@udel.edu

*Corrresponding author: partem@udel.edu

in both artificial and biological swarms.

In this paper, we propose a methodology to solve the problem of identifying the leader in a swarm governed by leader-follower control framework using only the macroscopic view of the entire swarm. A methodology that combines clustering and probability methods is used to analyze the spatial-temporal data of a swarm of robots to narrow down on a subset of the robots that includes the leader. The clustering parameters are optimized over multiple simulated behaviors. The results show that this automated system can narrow down a subset of agents (cluster) that includes the leader of the swarm with high accuracy. Applications of the method include automatic identification of leader and swarm dynamics in both artificial and biological swarms, which can lead to a better understanding of collective behaviors and predictions of future behaviors.

The rest of the paper is organized as follows: Section II introduces a modification to the Reynolds' behavioral model of swarms [19] in order to include a leader-follower framework. Then the proposed methodology is described, which involves clustering and probability methods in order to analyze the spatial-temporal data of the swarm and narrows down on a few agents which have a high probability of being the leader. We then formulate the optimization of the clustering parameters. Section III presents the results of the optimization method in simulated data, while Section IV concludes the paper by pointing out the contribution of the work and possible applications and directions for future work.

II. Problem Formulation

In this section, we first formulate the model our swarm will follow by introducing a modification to a well-known flocking model to include a leader agent. Then clustering and probability methods are introduced to analyze the resulting swarm motion as a spatial-temporal dataset. Finally, the parameters of the clustering method are optimized over a large set of simulations to identify the cluster the leader agent is in.

A. Modified Reynolds Flocking Model

All the simulated swarm behaviors discussed in this paper include a well-known and one of the mostly used flocking model in the literature, the Reynolds' model [19]. Since we are addressing the problem of leader identification in swarms, we modified the Reynolds model to incorporate a leader in the flock. The Reynolds' flocking model consists of three steering rules which each agent in the flock executes at each instance of time. The alignment and cohesion force components were modified to capture the behaviors of the agents attracted by the leader. We define the leader and agent dynamics separately below.

1) Agent Dynamics: Let $\mathbf{D}_i^t \in \mathbb{R}^2$ be the position of agent i at time t and $\mathbf{V}_i^t \in \mathbb{R}^2$ be the respective velocity. An agent i neighborhood N_i^r is the set of all agents other than agent i located within the sensing radius r of agent i. N_i^r={$j \in \mathbb{N}$ and $j \neq i \mid r_{ij} \leq r$ } where r_{ij} is the distance between agents i and j. All agents have the same sensing radius, however, the sensing radius of the leader is denoted by r_l. The indicator function $I_{N_l^{r_l}}(i)$ gives us the information if the agent i is in the leader neighborhood as shown below:

$$I_{N_l^{r_l}}(i) = \begin{cases} 1 & i \in N_l^{r_l} \\ 0 & i \notin N_l^{r_l} \end{cases} \tag{1}$$

We define the parameter $\mathbf{B}_{i,i\neq l}^t$ as below

$$\mathbf{B}_{i,i\neq l}^t = \sum_{\substack{j \in N_i^r \\ j\neq i,l}} \frac{\mathbf{V}_j^t + I_{N_l^{r_l}}(i).L\mathbf{V}_l^t}{|N_i^r| + I_{N_l^{r_l}}(i).L} \tag{2}$$

where L is the weight factor for the leader and $|N_i^r|$ is the number of agents in the neighbourhood of agent i. Using Eq. (2) we define the alignment force $\mathbf{F}_{alig,i,i\neq l}^t \in \mathbb{R}^2$ component acting on agent i at time step t as

$$\mathbf{F}_{alig,i,i\neq l}^t = \begin{cases} 0 & N_i^r = \{\} \text{ and } I_{N_l^{r_l}}(i) = 0 \\ \frac{\mathbf{B}_{i,i\neq l}^t}{\|\mathbf{B}_{i,i\neq l}^t\|} v_{max} - \mathbf{V}_i^t & \text{otherwise} \end{cases} \tag{3}$$

where $v_{max} = 2$ is the maximum speed of each agent empirically tuned. The alignment force $\mathbf{F}_{alig,i,i\neq l}^t$ ensures that the agent i steers towards the weighted average velocity of neighboring agents including the leader, if the agent i belongs to the leader neighborhood.

Furthermore, we define $\mathbf{P}_{i,i\neq l}^t$ and cohesion force $\mathbf{F}_{coh,i,i\neq l}^t \in \mathbb{R}^2$, acting on agent i at time step t, such that the agent i steers towards the weighted average position of the neighboring agents including the leader, if the agent i belongs to the leader neighborhood:

$$\mathbf{P}_{i,i\neq l}^t = \sum_{\substack{j \in N_i^r \\ j\neq i,l}} \frac{\mathbf{D}_j^t + I_{N_l^{r_l}}(i).L\mathbf{D}_l^t}{|N_i^r| + I_{N_l^{r_l}}(i).L} - D_i^t \tag{4}$$

$$\mathbf{F}_{coh,i,i\neq l}^t = \begin{cases} 0 & N_i^r = \{\} \text{ and } I_{N_l^{r_l}}(i) = 0 \\ \frac{\mathbf{P}_{i,i\neq l}^t}{\|\mathbf{P}_{i,i\neq l}^t\|} v_{max} - \mathbf{V}_i^t & \text{otherwise} \end{cases} \tag{5}$$

The separation behaviour is used to prevent collisions between the agents. The separation force $\mathbf{F}_{sep,i,i\neq l}^t \in \mathbb{R}^2$ acting on agent i at time step t is defined as

$$\mathbf{F}_{sep,i,i\neq l}^t = \begin{cases} 0 & N_i^r = \{\} \\ \frac{\mathbf{C}_{i,i\neq l}^t}{\|\mathbf{C}_{i,i\neq l}^t\|} v_{max} - \mathbf{V}_i^t & \text{otherwise} \end{cases} \tag{6}$$

where,

$$\mathbf{C}_{i,i\neq l}^t = \sum_{\substack{j \in N_i^r \\ j\neq i,l}} \frac{\mathbf{D}_i^t - \mathbf{D}_j^t}{\|\mathbf{D}_i^t - \mathbf{D}_j^t\|^2}. \tag{7}$$

Consequently, we combine equations (3), (5) and (6) to obtain the net acceleration $\mathbf{A}_{i,i\neq l}^t \in \mathbb{R}^2$ acting on agent i at time step t

$$\mathbf{A}_{i,i\neq l}^t = \left(w_1 \frac{\mathbf{F}_{alig,i}^t}{\|\mathbf{F}_{alig,i}^t\|} + w_2 \frac{\mathbf{F}_{coh,i}^t}{\|\mathbf{F}_{coh,i}^t\|} + w_3 \frac{\mathbf{F}_{sep,i}^t}{\|\mathbf{F}_{sep,i}^t\|} \right) a_{max} \tag{8}$$

where $w_1 = 1, w_2 = 0.2, w_3 = 1$ are tuned weights, and $a_{max} = 0.5$ is the maximum magnitude of acceleration of the agents empirically tuned.

2) Leader Dynamics: To simulate a leader-follower behavior we assign the following characteristics to the leader:

- The leader is given a fixed trajectory,
- other members/agents do not exert force on the leader,
- the leader has a bigger neighborhood radius than other members,
- while agents communicate with each other and follow rules of the flocking algorithm, the leader's position and velocity are given larger weights in finding cohesion and alignment forces for other agent(s) in the leader's neighborhood.

In the simulations, we decided to define a sinusoidal motion for the leader, and hence, its acceleration is given by

$$\mathbf{A}_l^t = \begin{bmatrix} 0 \\ 300\sin(2\pi t/400) \end{bmatrix} \tag{9}$$

The choice of sinusoidal motion profile is free and it does not interfere with the proposed algorithm, since the algorithm is not aware of any pre-prescribed motion profiles for the leader.

Finally, for all agents, we define unconstrained velocity $\mathbf{q}_i^{t+1}$ such that

$$\mathbf{q}_i^{t+1} = \mathbf{V}_i^t + \mathbf{A}_i^t \Delta t \tag{10}$$

where $\Delta t = 1$. Consequently, the velocity $\mathbf{V}_i^{t+1}$ and position $\mathbf{D}_i^{t+1}$ of agent i at time step $t+1$ are updated as follows:

$$\mathbf{V}_i^{t+1} = \frac{\mathbf{q}_i^{t+1}}{\left\|\mathbf{q}_i^{t+1}\right\|} v_{max} \tag{11}$$

$$\mathbf{D}_i^{t+1} = \mathbf{D}_i^t + \mathbf{V}_i^{t+1} \tag{12}$$

B. Clustering Method

The above model creates a spatial-temporal dataset of a swarm of agents that interact with each other. Given only the position of the agents with respect to time, we expect the agents to form clusters. To identify clusters in the dataset, the DBSCAN method [17] is a natural choice since it is efficient in clustering arbitrarily shaped clusters. The application of the DBSCAN is based a radius value ϵ and a value n_p for the minimum number of points that should be contained within the ϵ radius. These two parameters together determine the density of clusters. For convenience, we define the parameter set $\theta = \{\epsilon, n_p\}$.

Let "clustering" ζ_t be a partitioning of spatial-temporal data of agents in a swarm, achieved by DBSCAN method, at time step $t = \{1, 2, \ldots\}$. ζ_t contains clusters $C_1(t)$, $C_2(t)$, ... , $C_{N_t}(t)$. Let $\mathcal{N}_t = 1_t, 2_t, .., N_t$. Note that $C_m(t) \cap C_n(t) = \emptyset$ where $m, n \in \mathcal{N}_t$.

C. Probability Method for Leader Identification

We identify the cluster containing the leader using the following probability method formulation. Let $P_i(t)$ be the probability of each agent belonging in Cluster $C_i(t)$ at time step t to be the leader, where $i \in \mathcal{N}_t$. Let $X_i(t)$ be the total number of agents in Cluster $C_i(t)$ at time step t. Then, the probability of Cluster $C_i(t)$ containing the leader at time step t is given by $P_{C_i}(t) = P_i(t)X_i(t)$. The initial probability P_i(t=1) is considered proportional to the cluster size and defined as

$$P_i(1) = kX_i(1) \tag{13}$$

where k is a constant. The sum of probability of all clusters containing the leader, at any time step, should be equal to 1. Hence, we have

$$\sum_{i=1_t}^{N_t} P_i(t)X_i(t) = 1 \tag{14}$$

Combining equations (13) and (14) we get

$$k = \frac{1}{\sum_{i=1_t}^{N_t} X_i^2(1)} \tag{15}$$

Substituting the expression of k obtained from above in equation (13), the probability per agent in Cluster i at $t = 1$ is obtained as

$$P_i(1) = \frac{X_i(1)}{\sum_{i=1_t}^{N_t} X_i^2(1)} \tag{16}$$

As the simulation progresses, the agents shuffle and make new clusters. Fig. 1 depicts one such instance of agent shuffling. To assign the probability to the new cluster, we use the agent's cluster history. Each agent in the new cluster contributes to the new cluster's probability with information coming from its previous cluster. Let $d_{ij}(t)$ be the number of agents moved from Cluster $C_i(t)$, at time step t, to Cluster $C_j(t+1)$ formed at time step $t+1$ where $i \in \mathcal{N}_t$ and $j \in \mathcal{N}_{t+1}$. We have,

$$d_{ij}(t) = C_i(t) \cap C_j(t+1) \tag{17}$$

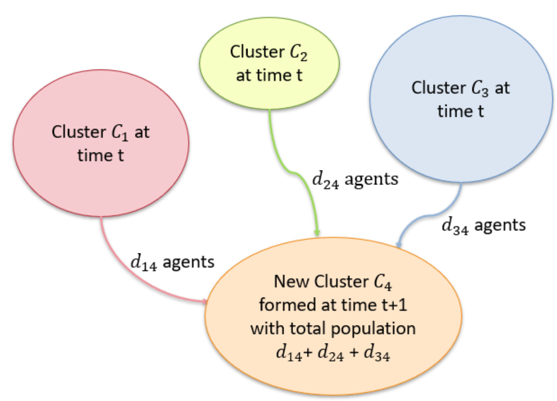

Fig. 1. New cluster formation with shuffling of agents from previous clusters.

The probability per agent in Cluster $C_j(t+1)$ at time step $t+1$ is defined as

$$P_j(t+1) = \frac{\sum_{i=1_t}^{N_t} d_{ij}(t)P_i(t)}{\sum_{i=1_t}^{N_t} d_{ij}(t)} \tag{18}$$

In other words, the probability of the agents of the new cluster described by equation (18) is essentially the weighted average of the probability of those agents in their previous cluster. The above rule is applied at each step to calculate each agent's probability to belong in a cluster that includes the leader, while it allows cluster merging and splitting.

D. Parameter Optimization

The above method is used to calculate the probability of each Cluster $C_i(t)$ containing the leader at time step t, given by $P_{C_i}(t) = P_i(t)X_i(t)$. For reference, we will denote the methods previously described in subsection B (clustering) and C (probability calculation) as methods B and C respectively.

The modified Reynolds flocking model is used to generate 100 different simulations of swarms of 100 agents in two-dimensional space. Each simulation is executed for 1000 time steps. Each agent is given random initial conditions. The parameters used are $L = 500$, $r_l = 60$ units, $r = 30$ units, $T_{final} = 1000$ frames. A window size of 800 units by 400 units is used for the simulation. To restrict the agents in the window, agents follow an elastic collision when they hit the boundary of the window.

The resulted data were provided to method B followed by method C above. The cluster with the highest probability of containing the leader at the final time step from each simulation is considered. We consider that the combined method (methods B and C combined) gave us the correct result if that cluster actually contained the leader. Then, we build an optimization method to find the optimum parameter set $\theta = \{\epsilon, n_p\}$ that maximizes the correctness of the leader prediction over all simulations.

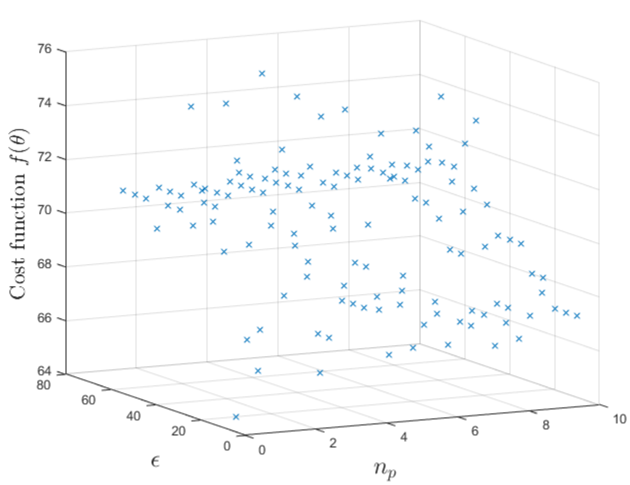

Fig. 2. Cost function values over combinations of the parameters ϵ and n_p.

We define a cost function for the optimization problem. Let $\mathcal{M}$ be the range of the cost function, where $\mathcal{M} = \{0, 1, 2, .., 100\}$. $\theta \in \mathbb{Z}^2$ is the optimization variable and $f : \mathbb{Z}^2 \to \mathcal{M}$, is the cost function. The cost function maps the parameter set θ to the number of correct results in the 100 simulations. The bounds on ϵ and n_p are chosen empirically for the following optimization problem.

$$\begin{aligned} \underset{\epsilon,\, n_p}{\text{maximize}} \quad & f(\theta) \\ \text{subject to} \quad & 1 \leq \epsilon \leq 80, \\ & 1 \leq n_p \leq 10, \\ & \epsilon, n_p \in \mathbb{Z} \end{aligned}$$

III. Simulations and Results

All algorithms were implemented and executed in MATLAB™ version R2020b (The MathWorks, Natick, MA USA). Simulation results were computed on a PC running a 64 bit Windows 10 operating system with an Intel® Core™ i7-9700 CPU @3.00GHz x 8 and 16GB of RAM.

We initially solved the problem heuristically, by testing all possible combinations of the integer constraints and parameters. The optimization took about 20 hours to be solved with parallel computation. After solving the optimization problem we got that the cost function attains the maximum value of 76, which translates to 76 correct predictions out of 100. This value corresponded to the optimized parameters $\epsilon = 40$ and $n_p = 3$. The cost function values over the set of parameters is shown in Fig. 2.

We also solved the same problem using Particle Swarm Optimization (PSO) method [20]. We used 20 iterations, population size= 5, $c_1 = 4$, $c_2 = 4$, and $w = 0.99$ for the simulation. The results using the PSO were close to our optimized set calculated before, i.e., $\epsilon = 39$ and $n_p = 3$ for which the cost function $f(\theta)$ had also a maximum value of 76.

Snapshots of one of the simulations are shown in Fig. 3. The snapshots show all the agent positions at specific time instances, the clusters they belong to, while the cluster with the highest probability of containing the leader is represented by a convex hull. The leader is also noted. As the simulation progresses, the cluster containing the leader achieves the highest probability, as demonstrated in Fig. 3. The video of the simulation can be found at [21].

Fig. 4 shows the probability of each of the clusters containing the leader with respect to time for the same simulation presented in Fig. 3. As it is shown, Cluster 1 starts with the highest probability, as time progresses Cluster 4 and Cluster 2 also start attracting agents because of the proximity of agents to those Clusters. We observe that for a short duration Cluster 4 and Cluster 2 indicate the highest probability of containing the cluster. Also, the agents in Cluster 3-9 over the period of time join other clusters and are left with no agents. We observe that the probability associated with those clusters goes to 0. In this simulation, given sufficient time, the actual cluster containing the leader in the simulation indeed achieves the highest probability.

IV. Conclusions

In this paper, we considered the problem of identifying the leader in a swarm of robots. We formulated the problem using the DBSCAN method for the spatial-temporal dataset to

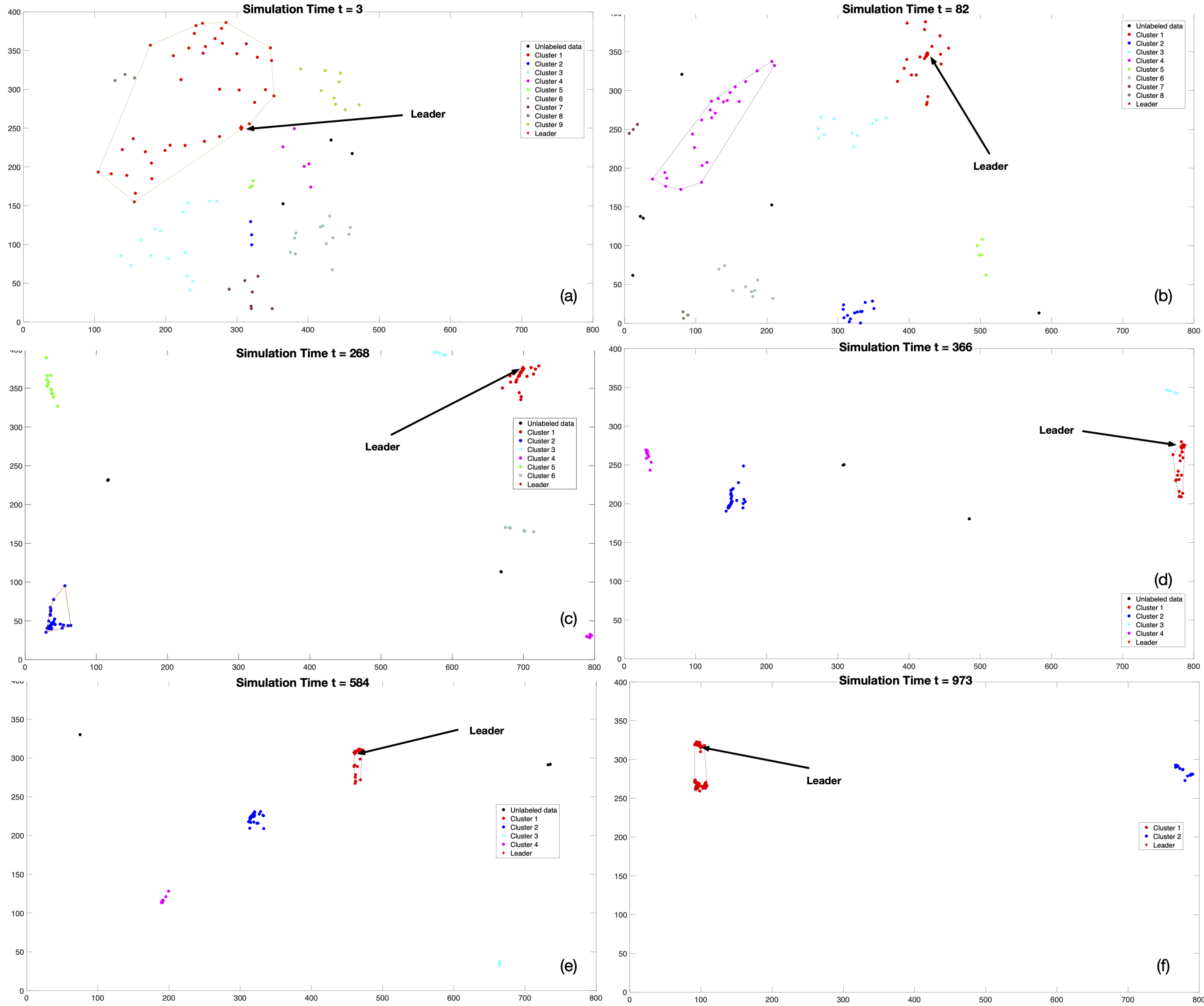

Fig. 3. Consecutive snapshots of spatial-temporal data of 100 agents with clustering parameters of $\epsilon = 40$ and $n_p = 3$. Snapshots at simulation times $t = 3, 82, 268, 366, 584, 973$ are shown in sub-figures (a)-(f) respectively. In each snapshot the leader is shown, while the cluster in a convex hull represents the cluster with the highest probability of containing the leader. (a) Although the leader is included in the most probable cluster, the agents are still widespread. (b-c) The cluster with the highest probability does not include the leader. (d-f) The clusters are getting less in number and the leader is always included in the most probable cluster.

define clusters of agents, and proposed a probability method to predict the cluster containing the leader. We optimized the parameters of our method to maximize its accuracy over a wide range of simulations, specifically 100 different simulations of agents. The agent and leader dynamics were governed by a modified Reynolds flocking model.

Results demonstrated that our proposed method could predict the cluster containing the leader with 76% accuracy. We observed that the cases where the unlabeled data, i.e., agents which do not belong to any cluster, had more than 13% of all agents initially, contributed to 20.8% of incorrect results at the end of the simulation ($t = 1000$). We also observed that 45.8% of the incorrect results had an initial cluster containing more than 60% of all agents.

The contribution of this work can be found in an optimized methods for identifying a leader agent in a swarm of agents governed by a modified Reynolds flocking model. This problems has not been sufficiently explored in the past and the current work provides a foundation for addressing it. Since the flocking model can be applied to both biological and artificial swarms, the proposed method can be considered as a very useful tool for identifying behaviors of swarms, that could be used to make future predictions of their motions or better understand their behavior.

Future developments include further analysis of the method in swarms with multiple leaders. Moreover, our plan is to introduce a human observer in the loop that could further improve the combined human-machine predictions.

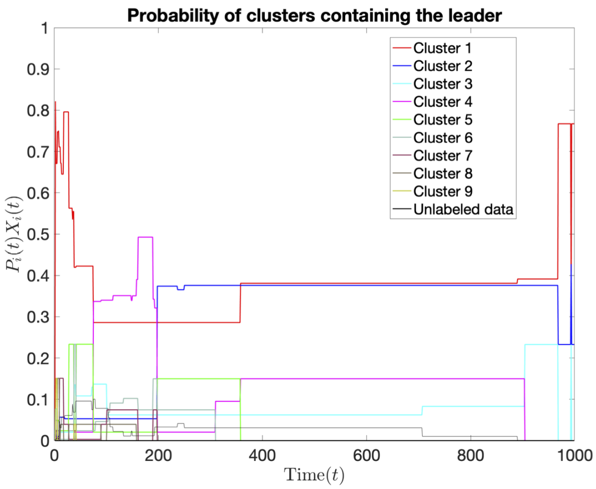

Fig. 4. Probability of clusters containing the leader with respect to time.

REFERENCES

[1] F. Zou, H. Jones, D. Jiang, T.-M. Lee, A. Martínez, K. Sieving, M. Zhang, Q. Zhang, E. Goodale, *et al.*, "The conservation implications of mixed-species flocking in terrestrial birds, a globally-distributed species interaction network," *Biological Conservation*, vol. 224, pp. 267–276, 2018.

[2] A. Fünfhaus, J. Göbel, J. Ebeling, H. Knispel, E. Garcia-Gonzalez, and E. Genersch, "Swarming motility and biofilm formation of paenibacillus larvae, the etiological agent of american foulbrood of honey bees (apis mellifera)," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[3] M. Egerstedt, T. Balch, F. Dellaert, F. Delmotte, and Z. Khan, "What are the ants doing? vision-based tracking and reconstruction of control programs," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 4182–4187.

[4] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "Nasa's swarm missions: The challenge of building autonomous software," *IT professional*, vol. 6, no. 5, pp. 47–52, 2004.

[5] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE transactions on robotics and automation*, vol. 14, no. 6, pp. 926–939, 1998.

[6] X. Wang, P. Lee, A. Ray, and S. Phopa, "A behavior-based collaborative multi-agent system," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, vol. 5. IEEE, 2003, pp. 4242–4248.

[7] H. Du, W. Zhu, G. Wen, Z. Duan, and J. Lü, "Distributed formation control of multiple quadrotor aircraft based on nonsmooth consensus algorithms," *IEEE transactions on cybernetics*, vol. 49, no. 1, pp. 342–353, 2017.

[8] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on control systems technology*, vol. 9, no. 6, pp. 777–790, 2001.

[9] H.-H. Hsu and A. Liu, "Multiple teams for mobile robot formation control," in *Proceedings of the 2004 IEEE International Symposium on Intelligent Control, 2004.* IEEE, 2004, pp. 168–173.

[10] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling swarms of robots using interpolated implicit functions," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2487–2492.

[11] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 73–80.

[12] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory & Applications*, vol. 1, no. 2, pp. 505–512, 2007.

[13] P. Walker, S. A. Amraii, M. Lewis, N. Chakraborty, and K. Sycara, "Human control of leader-based swarms," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 2712–2717.

[14] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2014.

[15] P. Kalnis, N. Mamoulis, and S. Bakiras, "On discovering moving clusters in spatio-temporal data," in *International symposium on spatial and temporal databases*. Springer, 2005, pp. 364–381.

[16] C. S. Jensen, D. Lin, and B. C. Ooi, "Continuous clustering of moving objects," *IEEE transactions on knowledge and data engineering*, vol. 19, no. 9, pp. 1161–1174, 2007.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *Int. Conf. Knowledge Discovery and Data Mining*, vol. 240, 1996, p. 6.

[18] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[19] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 69–73.

[21] (2021) Video simulation. [Online]. Available: https://bit.ly/3jtcJeM