A Graph-based Approach for Shepherding Swarms with Limited Sensing Range

Reem E. Mohamed*, Saber Elsayed*, Robert Hunjet†, Hussein Abbass*
* School of Engineering and Information Technology, University of New South Wales, Canberra ACT, Australia
Emails: r.sherif@student.adfa.edu.au, r.emk.sherif@gmail.com
Emails: {s.elsayed, h.abbass}@unsw.edu.au
†Defence Science and Technology Group, Edinburgh SA, Australia
Email: Robert.Hunjet@dst.defence.gov.au

***Abstract*—Within applications of swarm control, failing to maintain cohesion amongst the agents may lead to mission failure. We study the effect of limited sensing range of swarming agents when guided by a shepherd. A connectivity-aware approach is proposed to enhance the cohesion of the swarm. We combine a graph-based model of the flock with particle swarm optimization (assisted by DBSCAN) to improve the shepherd's performance in herding sheep (swarm members). The approach is evaluated using multiple initial swarm configurations. Simulation results on swarm sizes of 50 and 100 agents show up to 50% reduction in the task completion time and an average improvement of 25% in the success rate for low density sheep initialization scenarios and competitive results for the higher density scenarios.**

***Index Terms*—Shepherding, sensing range, particle swarm optimization, swarm control, sensing induced graph, DBSCAN**

I. Introduction

Guiding a flock of agents from their initial location to a specific destination through the use of an external agent has many important application areas, such as autonomous collection of oil spilt from oil tankers [1], inhibiting birds from approaching airports [2], guiding crowds of people away from dangerous areas [3], and guiding tours via neuron migration approaches [4]. This problem class is known as the shepherding problems, which was introduced in [5] in the context of animal behavior. The shepherding problem describes a task of a single or multiple agents, acting as sheepdog(s), to guide a large number of agents, modeled as swarms, interacting through the applied force by the shepherd to move from their initial location to a goal locations [4].

In the context of robotic swarms, the flocking of agents with connectivity preservation has been intensively studied by finding control laws for agents to improve their collective motion [6]. The work was improved by deploying particle swarm optimization (PSO) [7] in [8]. Following this approach, connectivity preservation in shepherding is considered in the N-wavefront algorithm introduced in [9] and extended in [2], where finding the feasible herder trajectory that minimizes dispersion was studied. Unlike shepherding, which aims to guide the flock to a specific destination, this work aimed to divert the flock from a specific location to protect an airport from a flock of birds with negligence to its final destination.

The shepherding problem has been tackled using geometric [10] and heuristic [11] methods that have been verified through numerical and practical experiments. It has been extended by deploying differential evolution (DE) based path planning method to minimize the dispersion of sheep in a cluttered environment [12]. Moreover, intensive system analysis and detailed design of a sheepdog application in human swarm control was introduced in [13]. In that work, the workflow of the agents was described showing the importance of the sensorial information in the decision making of the controlling agent. Moreover, the complexity of the shepherding problem has been intensively studied and analysed in [14]. The study demonstrated the effect of different settings on the feasibility of task completion in a limited time. However, the effect of physical constraints on the controlled agents remains open for investigation.

To the best of our knowledge, the existing work mentioned above does not consider realistic assumptions about the sensing capabilities of agents. Generally, global knowledge of the flock is assumed in the existing literature. In a robotic/physical implementation, a limited sensing range may deteriorate the performance of existing approaches.

With the above identified gap, we approach the problem in this paper from a networking perspective; modelling the flock as a graph and seeking to maintain its connectivity. We then select a subset of sheep to directly influence the flock, finding a near optimal goal point for the shepherd to effectively guide all the sheep to, whilst avoiding dispersion, in a constrained-time environment. The shepherd is also made "connectivity-aware," maintaining an appropriate distance from the flock to avoid dispersing the sheep. This process is repeated when the shepherd reaches the estimated herding point.

The proposed method, Swarm Optimal Herding Point (SOHP), has been tested on multiple scenarios in simulation and has been found to exhibit a higher success rate and a reduced mission completion time compared to existing approaches.

The contributions of this paper can be summarised as follows:

- Defining the flock in terms of a unit disk graph (UDG) according to the sheep sensing range to represent the sensing induced graph among homogeneous agents with omni-directional sensors.

- Developing a geometric method to specify a subset of sheep to be influenced by the shepherd
- Combining geometric rules with meta-heuristic optimization method aided by clustering method to find the near optimal herding point whilst considering flock connectivity.

The rest of this paper is organised as follows. The shepherding problem formulation is discussed in Section II. In Section III, the proposed swarm optimization-based method is explained in details. In Section IV, the results of the proposed method in different initialization scenarios are discussed. Finally, conclusions are drawn in Section V along with a discussion of future research directions.

II. Problem Formulation

In this section, we describe the general shepherding problem and the effect of limited sensing range.

We use π to refer to a sheep, β to refer to a shepherd, $R_{\pi\pi}$ to the sheep sensing range of their peers, R_{π} to each sheep safe range that makes it avoid collision with its peers, $R_{\pi\beta}$ the sheep sensing range of the shepherd, N is the set of sheep to be herded and N^t are the subset of sheep yet to reach the goal area at time t where $|N|$ refers to the cardinality of N. However, for simplicity, we use the set name to refer to its size or add the word set before the notation otherwise. Moreover, P_i^t is the position at time t in two dimensions x and y for agent i, $dir(p,q)$ is the unit vector of the direction between the points p and q in two-dimensional space, and $d(p,q)$ is the Euclidean distance between p and q in two dimensional space.

Shepherding is the process of collecting and driving a group of agents, like sheep, to a home location by using another agent or group of agents, sheepdogs [11]. A cohesive flock is formed via the application of different force vectors, representing attraction and repulsion. Each sheep agent is attracted to the Local Center of Mass (LCM) of the other sheep within its $R_{\pi\pi}$. This LCM represents its goal location at time t, which is calculated at each time step. Additionally, the sheep repel from its neighbors within its safe radius R_{π} to avoid collision and from the shepherd within its agitation range $R_{\pi\beta}$, where $R_{\pi} < R_{\pi\pi} < R_{\pi\beta}$. Thus, a sheep agent is influenced by the sum of the attracting force towards its LCM, the sum of repulsive forces from its neighbouring sheep within its R_{π} and the sum of repulsive forces from the shepherds. The set of sheep within R_{π} for any π is a repulsive set n_{rep}. Assuming Additive white Gaussian noise (AWGN) to the final direction and considering the effect of inertia on the sheep, the resulting force vector applied to each sheep is given in Eq.1 as described in [11] where all agents are assumed to be particles with unit mass. Thus, the acceleration of each particle is the sum of all forces due to $F = ma$.

$$\mathrm{F}^t_{total} = W_{\pi\Lambda} dir(P^t, \Lambda^t) + W_{\pi\pi} \sum_{i\in n_{rep}} dir(P^t, P_i^t) + W_{\pi\beta} dir(P^t, P_{\beta}^t) + W_{e\pi_i} dir(P^t, rand) + W_{\pi_v} dir(P^t, P^{t-1}) \quad (1)$$

where $W_{\pi\pi}$ is the π repulsion strength from π, $W_{\pi\beta}$ the π repulsion strength from β, $W_{\pi\Lambda}$ the π attraction strength to Λ, $W_{\pi v}$ the Strength of π previous direction, $W_{e\pi}$ the strength of sheep π angular noise, and $rand$ is a random position representing a jittering effect.

The effect of these competing forces on sheep motion and their lack of awareness of their goal location impacts complexity in the shepherding task. The shepherd must select appropriate intermediate goal points to guide the sheep in the desired direction whilst ensuring the flock does not disperse, noting that the presence of the shepherd itself in close proximity to the sheep can cause dispersion. Minimum cost paths, such as those generated through traditional path planning approaches may not be sufficient to satisfy these requirements in the shepherding task.

To ensure the success of shepherding in $T < \infty$ time, the approach should be able to decrease displacement between the sheep and the goal at each time step $t < T$, and ensure the cohesion of the sheep throughout the task time.

III. Swarm Optimal Herding Point

This section describes the proposed method that is inspired by the centroid push-based shepherding model [11], i.e. the centroid of the flock is progressively moving closer to the goal, whilst maintaining flock cohesion. Swarm Optimal Herding Point (SOHP) is designed to improve the performance of a single shepherd in overcoming the sheep dispersion due to limited sheep sensing range and comprises two main steps applied sequentially. The first is to select the shepherd behaviour and subset of sheep to influence, and the second is to estimate the near-optimal herding point. We discuss these steps below.

A. Flock Subset Selection

Our approach considers the set of sheep with a limited $R_{\pi\pi}$ as a dynamic Unit Disk Graph (UDG) that changes every time step t due to sheep movement. Throughout this paper the graph $G = (N, \mathcal{E})$ is an undirected, connected graph with a vertex set $N = 1, 2, ...|N|$ and edge set $\mathcal{E}$ with length bounded by $R_{\pi\pi}$. $\Delta(i)$ is the degree of vertex i that represents the number of its neighbors, i.e. those within its sensing range $R_{\pi\pi}$. The set of sheep is a connected component (cc) if all sheep within it are connected either directly or via other sheep in the sensing induced graph. A *bridge* is an edge that if removed, it fragments a connected component.

This step identifies the subset of the flock to be influenced by the shepherd. This subset has $|N_{subset}| < |N^t|$ sheep, where $|N^t| = |N|$ at $t = 0$. Inclusion in the subset is dependent on the ordered distances between all the sheep and their goal location (and the global centre of mass (GCM) if needed, depending on the sheep connectivity) as described in Algorithm1. Sheep connectivity metrics govern the decision as to whether sheep are collected (the flock made more cohesive), or driven to the goal. A UDG approach is used to create a sensing induced graph based on sheep locations and limited sensing range. The minimum node degree, number of bridges and connected components are measured (step:1-2) so that the shepherd guides the sheep while maintaining the UDG connectivity during the collecting behaviour.

The next steps (step:3-6) are for the selection of proper shepherd behaviour, i.e., (1) driving the sheep to the goal without considering their connectivity, or (2) collecting the sheep by considering their connectivity, whilst guiding them to the goal. The shepherd stops if it is within a predefined D_{stop} distance from any π while adopting any of the two behaviours, similar to [11].

The driving behaviour is selected if and only if all the following three conditions are satisfied in the sensing induced graph at time t: (a) the minimum node degree in G is more than a threshold value Δ_{thresh}, (b) the sheep N^t form one connected component n^t_{cc}, and (c) the graph has no bridges $n_{bridges} = 0$. In that case, the subset of the flock to influence the set $|N^t_{subset}|$ is the cc that has the furthest sheep from the goal. Summarising, if the flock is well connected (a single cc in the absence of bridges) and the minimum node degree of the sheep is above a specific threshold, the flock is assumed to be cohesive and is driven towards the goal.

When the conditions for driving are not satisfied, subset selection for the collecting behaviour is performed considering the distance to the sheep GCM. The collecting behaviour is designed to increase the level of connectivity in the overall sheep graph. In this case, inclusion in N^t_{subset} is based on sheep distance either from home or from the GCM, the values of which are stored for each sheep in two lists D^t_1 and D^t_2, respectively. The sheep id of the sheep with maximum D^t_1, if the shepherd is driving, or maximum in both D^t_1 and D^t_2 if the shepherd is collecting, is stored as 'node'. The connected component that has the 'node' sheep is the selected subset (step:7-14). However a maximum subset size of $|N_{subset}|_{max}$ is imposed to provide the shepherd with a manageable number of sheep to influence and guide. Thus, only $|N_{subset}|_{max}$ of the closest sheep to the 'node' form N_{subset}, if the size of the selected connected component exceeds $|N_{subset}|_{max}$ (step:14-18).

Algorithm 1: SubsetSelection($P_{i\in N^t}$)

Input : $P_{i\in N^t}$
Output : N^t_{target},Behaviour
1 Create G^t
2 Find n_{cc}, Δ_{min}, and $n_{bridges}$ at time t
3 **if** $n_{cc} > 1$ *or* $n_{bridges} > 1$ *or* $\Delta_{min} < \Delta_{thresh}$ **then**
4   Behaviour='Collect'
5 **else**
6   Behaviour='Drive'
7 Measure D^t_1
8 **if** *Behaviour='Collect'* **then**
9   Measure D^t_2
10   node=id of sheep at max(D^t_1,D^t_2)
11 **else**
12   sort D^t_1 Descending
13   node=id of sheep at max(D^t_1)
14 cc*= the connected component of node
15 **if** $|cc^*| > |N_{subset}|_{max}$ **then**
16     $N^t_{subset} \leftarrow thenearest$—N$_{subset}|_{max}$ to node
17 **else**
18   N^t_{subset}←cc*

B. Finding the Near Optimal Herding Point

With the shepherd behaviour and subset of the flock to influence defined, we offer an approach to enhance connectivity of the graph whilst reducing distance to the goal as outlined in Algorithm 2. This is achieved through influencing the subset simultaneously towards their LCM and home location. To do this, we select for N^t_{subset} a point that is on the line between its LCM and the home location, where the sheep achieve cohesion since their cohesive force (flock) and agitation (shepherd) force are closely aligned.

With this approach, the problem of finding the steering direction is modeled as a single objective function F with a weighted sum of two average Euclidean distances between N^t_{subset} and two goal points as in Eq. 2. The first (f_1) measures the average distance between the sheep in the N^t_{subset} and the home location H as the goal point for this function as in Eq. 3. The second (f_2) measures the average distance between the N^t_{target} sheep and their LCM as the goal point for this function. Note that f_2 affects the strength of the connectivity among N^t_{subset} Eq. 4. The minimization of F aims to find the new locations for the sheep that minimizes the distance between the two goals simultaneously. This goal point is defined by predefined weights that control how close it is to the home location and the LCM.

$$\min_{x_{i\in N^t_{subset}}, y_{i\in N^t_{subset}}} \quad F = w_1 f_1 + w_2 f_2 \tag{2}$$

where

$$\mathrm{f}_1 = \textstyle\sum_{i\in N^t_{subset}} \sqrt{(x_i - H_x)^2 + (y_i - H_y)^2}/|N^t_{subset}| \tag{3}$$

$$\mathrm{f}_2 = \textstyle\sum_{i\in N^t_{subset}} \sqrt{(x_i - LCM_x)^2 + (y_i - LCM_y)^2}/|N^t_{subset}| \tag{4}$$

Note, the objective function aims to find the straight line that links each sheep in N^t_{subset} with the location that is on the straight line between the home location and its LCM. If the shepherd behaviour is 'collect', The weights w_1 and w_2 are calculated at time t as a function of the sheep connectivity graph G^t metrics (see Eq. 5), otherwise, the weights are set to $w_1 = 1$ and $w_2 = 0$ since the summation of both weights is always one as in Eq. 6 (step: 1-4).

$$w_1 = \begin{cases} e^x/(e^x+1) & 'collect' \\ 1 & 'drive' \end{cases} \tag{5}$$

where $x = (\Delta_{avg} - (n_{bridges} + n_{cc})\Delta_{min})/\Delta_{avg}$

$$w_2 = 1 - w_1 \tag{6}$$

We employ PSO to solve the above optimization with N^t_{subset} as particles forming the initial solution. In each iteration, the velocity of each particle and the position of the particle are updated according to Eq. 7 and Eq. 8 respectively.

While the sheep step size is used to bound the solution space such that valid solutions only exist within k_{steps} steps away from each sheep initial location as in Eq. 9. Moreover, the number of iterations within the optimisation is calculated using Eq. 10.

$$\mathrm{v}_i(t+1) = wv_i(t) + c_1 r_1[x_i(t) - x_i(t)] + c_2 r_2[g(t) - x_i(t)] \quad (7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

$$ub = P_\pi^t + k_{steps}\delta_\pi, lb = P_\pi^t - k_{steps}\delta_\pi \quad (9)$$

$$n_{iter} = k_{iter} + k_{iter} * (n_{cc} * (|N| - |N_{subset}^t|)/|N|) \quad (10)$$

We use i as particle index,w as the inertia coefficient for local and global velocities that are accelerated by coefficients c_1 and c_2 respectively such that $0 \leq c_1, c_2 \leq 2$, $0 \leq r_1, r_2 \leq 1$ are generated random values, and k_{iter} is a system predefined constant for setting the minimum number of iterations.

The decision of the final steering direction is the direction of the line linking the initial location of each particle and the final solution. The final solutions are the final locations of each particle or sheep in N_{subset} that minimizes the distance between this sheep and the home and the LCM of N_{subset} simultaneously. These locations are used to get the direction that each sheep π took to get to that point dir_π using Eq.11 and these directions are stacked in dir_{subset} (step: 5-6)

$$dir_\pi = dir(P_\pi^{init}, P_\pi^{fin}) \quad (11)$$

we refer to the sheep initial position with P_π^{init} and its position after n_{iter} with P_π^{fin}.

This final steering direction that the shepherd guides N_{subset} to is determined according to one of the following three cases of the values in dir_{subset} (step: 7-14):

- All N_{target}^t agree on the same direction that is $(0,0)$; thus, the shepherd influences the LCM of N_{subset}^t towards GCM, if it is collecting, or guides the sheep GCM towards the goal, if it is driving.
- All the N_{subset}^t agree on the same direction that is not $(0,0)$; thus, this direction is the steering direction of the LCM of N_{subset}^t at timestep t.
- The N_{subset}^t have different directions; thus, the directions are clustered using DBSCAN [15] where its parameters υ and ϵ are the maximum separation between points in each cluster and minimum number of points in a cluster. The average direction of the cluster with the highest density of directions is the steering direction of the LCM of N_{target}^t at timestep t.

Algorithm 2: HerdingPointEstimation(N_{subset}^t,Behaviour)

Input : N_{target}^t,Behaviour
Output : H^t
1 **if** *Behaviour='Drive'* **then**
2     $w_1 = 1$ and $w_2 = 0$
3 **else**
4   set w_1, w_2, see eqs.(5,6)
5 Initialize solutions with locations of N_{subset}^t
6 Use PSO to minimize using Eq. 2) to get the best direction of each particle as a set dir_{subset}
7 **if** dir_{subset} *not converged* **then**
8   Set DBSCAN parameters; υ_{DBSCAN},d_{DBSCAN}
9   DBSCAN(dir_{subset})
10     $dir^* = mean(cluster_{max})$
11 **else**
12     $dir^* = any(dir_{subset})$
13 **if** $dir^* = (0,0)$ **then**
14     $dir^* = d(LCM_{N_{subset}}, H)$
15 $H^t \leftarrow$LCM$_{N_{subset}} - dir^* D_{herding}$

The herding point for the next time step H^t is $D_{herding}$ away from the LCM of the N_{target}^t in the opposite direction of the near optimal steering direction obtained dir^* (step: 15). The shepherd goes to H^t in the following time steps and repeats Algorithm 1 then Algorithm 2 when it reaches it, thereby generating the subsequent point. Note that the optimality of this point diminishes during the shepherd's traversal towards it due to the dynamic nature of the flock.

IV. Results

In this section, the simulation scenarios and results obtained are discussed.

A. Experimental setup

To test SOHP with limited sensing range of sheep, different environmental scenarios are generated. First, we initialize the sheep in a square of length l defined as a product of $R_{\pi\pi}$,$\sqrt{N}$, and a constant number k as formulated in Eq. 12

$$l = kR_{\pi\pi}\sqrt{N} \quad (12)$$

Thus, the initial density of sheep ρ is defined in Eq. 13

$$\rho = N/l^2 = N/(kR_{\pi\pi}\sqrt{N})^2 = 1/(kR_{\pi\pi})^2 \quad (13)$$

Note that $k > 0$ is a density factor because for constant $R_{\pi\pi}$ and N, it is inversely proportional to sheep initialization density

Second, we mathematically estimate the task time required for completing the shepherding task by considering it as a function of the system initial location parameters which include (a) goal location,(b) shepherd initial location, and (c) sheep initial locations and density. We approached this formulation by mapping each element in the equation used as the maximum task time in [11] Eq. 14 to another function in these initialization parameters. The first element becomes a function in the sheep initialization density and the second one is a function in sheep and shepherd initial displacements from home

$$T_{max} = 20N + 630 \quad (14)$$

For the first element $20N$ in Eq. 14, the constant 20 is replaced with a function in the sheep sensing range $R_{\pi\pi}$ and the density factor k. This is because k is inversely proportional to the sheep density at constant $R_{\pi\pi}$, and thus a longer herding time is needed in collecting the sheep as k increases. In that case, the task time is directly proportional to k. Thus, replacing a constant with a function of sheep initialization density provides a more flexible approach for evaluating the shepherd performance. The mapping of the first element is in Eq. 15

$$20N \rightarrow kR_{\pi\pi}N \quad (15)$$

The second element 630 in eq. 14 is replaced with the total time taken by a single shepherd initialized at $P^0_\beta$to reach its initial herding point H^0 in addition to the time taken by the furthest sheep from home P^0 to reach home H. This is done ideally by calculating the initial displacements between the shepherd and the initial herding point H^0 divided by the maximum shepherd step size δ_β. Additionally, the initial distance between the furthest sheep from home initialized at P^0_π and the home H is divided by the maximum sheep step size δ_π to estimate the time needed by this sheep to reach home. Thus, we roughly estimate the number of steps required to herd only one sheep initialized in the location of the furthest sheep from home. However, the herding task involves multiple sheep which implies longer time than this estimated time. Thus, we introduce a constant number $k_1 > 1$ to be multiplied by the sum of the aforementioned estimated travelling times to represent the distance tolerance in estimation. In this way, the second element in Eq. 14 is mapped to the estimated travel time as in Eq. 16

$$630 \rightarrow k_1(d(H^0, H)/\delta_\pi + d(H^0, P^0_\beta)/\delta_{\beta)} \quad (16)$$

Combining, the time affected by the density of the sheep initialization in Eq. 15 with the time required for herding one sheep in Eq. 16, we formulate the maximum task time in Eq. 17

$$\mathrm{T}_{max} = k_1(d(H(0), H)/\delta_\pi + d(H(0), P^0_\beta)/\delta_\beta) + kR_{\pi\pi}N \quad (17)$$

B. Comparative Analysis

To measure the proposed method's performance, we compare it with other centroid push models, namely, the Strombom model [11] and CADSHEEP [12]. We randomly initialize the flock at five different density factors with the simulation parameters described in Table I. The metrics used in the comparison are (a) success rate, which is the number of time the task succeeded before T_{max} time steps, (b) task time T which is the time required by the shepherd to make all the sheep reach the home are with radius R_H (c) percentage of sheep at home (% sheep at H), which is the average percentage of sheep that reach home at time $t = T_{max}$, (d) average node degree Δ_{avg} as defined in Section II, and (e) the number of connected components n_{cc} as defined in Section II. These metrics demonstrate the efficacy of the graph based approach over 25 runs on flock sizes 50 and 100 sheep as tabulated in Table II and Table III, respectively. Moreover, we measured the change in the average node degree and percentage of sheep reaching home throughout the task time (+/−) one standard deviation to show the effect on each method on the sheep graph and the progress of the shepherd, respectively.

TABLE I. SYSTEM ASSUMPTIONS

Parameter	Value
Number of sheep	$[50, 100]$
Maximum size of the subset of sheep $\lvert N_{subset}\rvert_{max}$	$N/2$
Environment length L	300
Shepherd initial location P^0_β	(L, L)
Distance tolerance constant k_1	2
Home Location (H)	(0,0)
Home radius R_H	50
Minimum initial sheep distance to H	$L/4$
Density factor (k)	[1/4,1/3,1/2,2/3,3/4]
Shepherd maximum step size δ_β	5
Sheep maximum step size δ_π	2
Sheep step size in grazing	0.05
Sheep sensing radius for shepherd $R_{\pi\beta}$	70
PSO: c1,c2,w,k_{iter}	1.5,2,0.3,21
Number of steps bounding PSO search k_{steps}	7
DBSCAN: υ,ϵ	$N^t/10, 0.5$
Agitation weight $W_{\pi\beta}$	1.9
Sheep collision avoidance radius R_π	3
Collision weight $W_{\pi\pi}$	1.5
Sheep sensing radius $R_{\pi\pi}$	15
Cohesion weight $W_{\pi\Lambda}$	1
Threshold node degree Δ_{thresh}	$N^t/2$
Herding distance $D_{herding}$	$R_{\pi\beta}/2$
Jittering weight $W_{e\pi}$	0.3
shepherd stopping distance D_{stop}	$3R_\pi$
Inertia weight $W_{\pi\upsilon}$	0.5

The results in Table II and Table III show that the proposed method improved the success for 50 sheep in the case of low initialization densities at k=2/3,3/4, with improvements at these densities and k=1/2 in the case of 100 sheep. The overall degradation in high-density cases is below 15% offset by the improvement in the lower 2 and 3 densities for 50 and 100 sheep respectively which is up to 50%. This validates the ability of the graph-based approach to improve the cohesive motion of a weekly connected flock. As one would expect in higher density cases, there is sufficient connectivity by default. The task time in the high-density initialization cases for 50 sheep is longer for the proposed model by an average of 15%, once again offset by an average 30% decrease in the lower density cases. Also shown is that the proposed method is able to drive a large percentage of the sheep to the goal in the case that the overall goal is not achieved.

The last two rows in both tables describe the sheep graph metrics used in the proposed method in behaviour decision and weighing the objective function for finding the herding

TABLE II. NUMERICAL RESULTS FOR 50 SHEEP

	Metric	Best Mean			Mean			Standard Deviation		
Metric	Model	Strombom	CADSHEEP	SOHP	Strombom	CADSHEEP	SOHP	Strombom	CADSHEEP	SOHP
Success Rate	1/4	100	100	100	100	100	100	0	0	0
	1/3	100	100	100	100	100	100	0	0	0
	1/2	100	100	100	32	**84**	70.83	0.47	0.37	0.46
	2/3	100	100	100	36	40	**83.33**	0.49	0.5	0.38
	3/4	100	100	100	40	40	**79.17**	0.5	0.5	0.41
Task Time T	1/4	**89.00**	**89.00**	104	**89.76**	90	107.33	0.72	0.91	1.99
	1/3	**91.00**	92.00	105	93.16	**92.84**	110.92	0.89	0.85	4.32
	1/2	**100.00**	101.00	118	520.6	437.44	**340.30**	27.02	149.85	182.70
	2/3	340.00	363.00	**256**	648.96	688.72	**459.42**	19.33	121.07	152.33
	3/4	378.00	424.00	**370**	706.08	777.04	**547.83**	20.9	133.87	151.34
% sheep at H	1/4	100	100	100	100	100	100	0	0	0
	1/3	100	100	100	100	100	100	0	0	0
	1/2	100	100	100	40.4	**89.6**	77	44.33	25.34	40.37
	2/3	100	100	100	45.92	73.44	**91.92**	44.85	26.90	23.94
	3/4	100	100	100	47.28	74.72	**79.42**	47.18	27.88	41.00
Δ_{avg}	1/4	35.73	**36.11**	34.40	**34.15**	34.04	32.43	0.86	1.08	0.80
	1/3	31.44	**31.71**	30.48	29.36	**29.52**	28.81	0.84	1.06	0.81
	1/2	**41.34**	22.11	32.17	**27.02**	16.42	21.19	8.98	4.10	4.87
	2/3	**44.58**	18.88	17.96	**19.32**	10.55	14.99	7.19	3.63	2.13
	3/4	**38.38**	17.31	20.15	**20.90**	9.49	15.93	6.66	3.49	2.60
n_{cc}	1/4	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0
	1/3	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0
	1/2	**1**	1.01	**1**	1.82	1.62	**1.60**	0.51	0.33	0.35
	2/3	1.37	1.49	**1.53**	2.63	**2.18**	**2.18**	0.45	0.29	0.45
	3/4	1.77	**1.61**	1.8	2.83	**2.61**	2.67	0.49	0.53	0.43

TABLE III. NUMERICAL RESULTS FOR 100 SHEEP

	Metric	Best Mean			Mean			Standard Deviation		
Metric	Model	Strombom	CADSHEEP	SOHP	Strombom	CADSHEEP	SOHP	Strombom	CADSHEEP	SOHP
Success Rate	1/4	100	100	100	100	100	100	0	0	0
	1/3	100	100	100	56	**88**	83.33	0.50	0.33	0.38
	1/2	100	100	100	40	60	**70.83**	0.50	0.5	0.46
	2/3	100	100	100	32	36	**62.5**	0.48	0.49	0.50
	3/4	100	100	100	12	16	**33.33**	0.33	0.37	0.48
Task time T	1/4	94	**93**	118	**94.6**	94.72	122.21	0.64	0.61	2.02
	1/3	99	**99**	125	433.76	329.40	**241.79**	301.07	224.59	221.20
	1/2	286	**226**	396	772.56	804.04	**711.21**	307.18	259.91	206.41
	2/3	**374**	488	461	1052.32	1194.08	**995.79**	386.51	249.17	260.15
	3/4	394	931	**248**	1372.44	1418.6	**1092.04**	268.77	154.11	488.50
% sheep at H	1/4	100	100	100	100	100	100	0	0	0
	1/3	100	100	100	61.6	92.6	**96.83**	45.89	21.03	9.30
	1/2	100	100	100	54.32	83.6	**84.5**	42.64	24.51	31.24
	2/3	100	100	100	42.36	70.92	**73.17**	45.09	31.20	39.55
	3/4	100	100	100	26.48	**70.6**	68.96	37.43	31.60	93.53
Δ_{avg}	1/4	57.87	**58.39**	51.22	56.16	**56.22**	48.75	1.04	1.35	1.50
	1/3	**77.5**	48.71	50.98	**49.94**	39.24	38.97	18.30	8.19	8.80
	1/2	**67.94**	36.48	39.55	**36.88**	22.94	27.38	12.95	8.90	5.50
	2/3	**81.46**	25.56	33.88	**40.18**	15.97	25.50	17.15	5.63	4.29
	3/4	**81.69**	26.46	27.79	**39.16**	14.12	16.05	21.22	4.70	5.59
n_{cc}	1/4	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0.007
	1/3	**1**	**1**	**1**	1.31	1.52	**1.22**	0.37	0.48	0.37
	1/2	**1.18**	1.79	1.95	2.48	2.30	**2.47**	0.42	0.39	0.30
	2/3	**2.06**	2.12	2.36	**2.80**	3.22	3.07	0.36	0.95	0004
	3/4	**2.24**	2.33	2.30	**3.02**	3.72	3.68	0.69	0.78	1.21

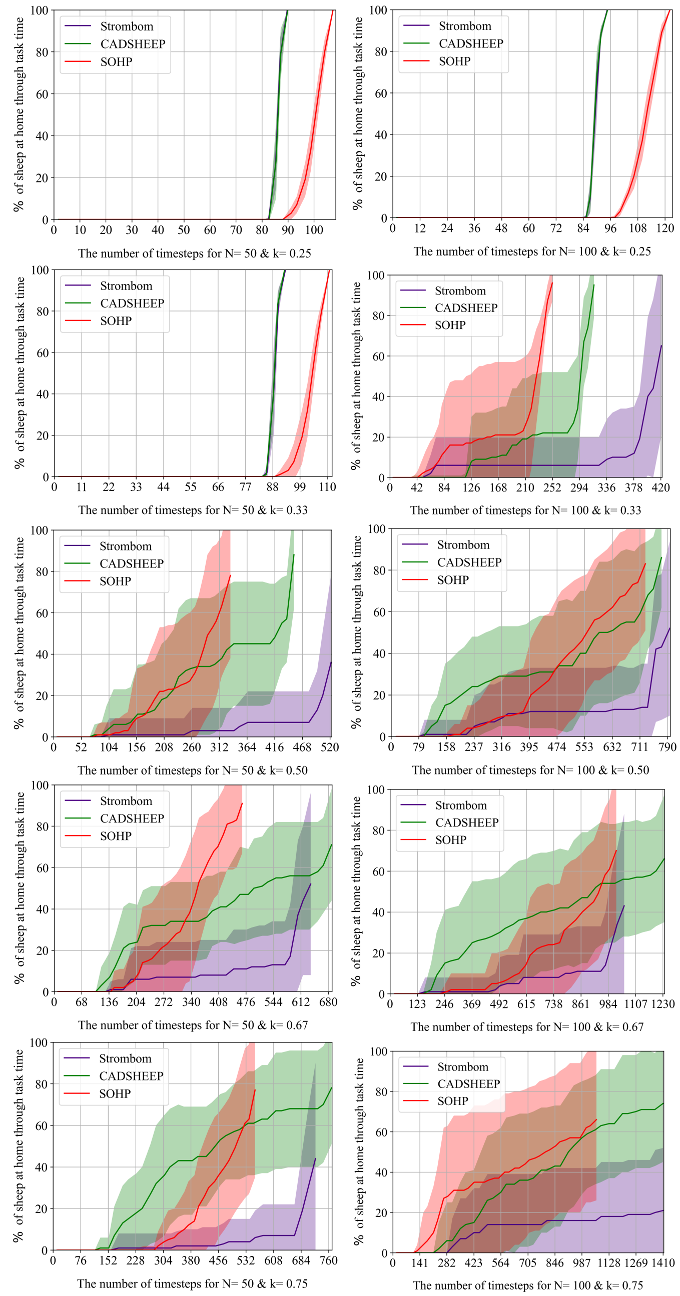

Fig. 1. The change in the average percentage of sheep reaching home $(+/-)$ one standard deviation for N=50 (left) and N=100 throughout the task time measured at different k=1/4,1/3,1/2,2/3,3/4 from top to bottom

point. These metrics are the average sheep node degree and the number of connected components throughout the task time. The results show that CADSHEEP and Strombom models maintain relatively high node degree and a low number of connected components in most of the cases despite they couldn't achieve lower task time or higher success rate in these initialization cases. This proved that the use of graph metrics in the proposed approach achieved its main objective, which is improving the shepherding task efficiency in terms of success rate and task time by effectively using the graph metrics without intensive concern on improving them. The bold font in the tables highlights the maximum success rate, node degree Δ_{avg} and percentage of sheep at home at the end of of task time obtained from Eq. 17, and the minimum task time, and number of connected components n_{cc} where these values are the most desirable in the shepherding problem.

The effectiveness of using the graph metrics by the proposed method is proven in the pattern of the change in Δ_{avg} throughout the average T of each method depicted in Figure 1 where each method has ends at a different time. The improvement of node degree by the proposed method through time is faster compared to its peers. Moreover, the curve is steady as the average node degree reached a fairly high value that enabled the shepherd to finish the task relatively faster than its peers with a higher success rate in low-density initialization cases. However, this relatively low node degree lead to slower task completion time compared to those of its peers when the sheep are initialized with high density. This proved the high efficiency of the proposed method in low-density initialization compared to its peers and fairly acceptable performance in high-density initialization instances. This is proven in the percentage of sheep arriving at home through the task time as shown in Figure 2, where in average the proposed method is about 15% slower in task completion time at k=1/4,1/3 for 50 sheep and k=1/4 for 100 sheep. While the pattern of sheep arrival to home is preserved as k increases, unlike its peers. This proves the suitability of the proposed method to handle dispersed sheep compared to its peers.

V. Conclusion and Future Work

In this paper, we presented a connectivity aware approach to shepherding where sheep agents have limited sensing range. This was done by modeling the flock as a UDG to quantify the sheep cohesion through graph metrics. Subsequently, PSO was used to find a near optimal herding point for the shepherd to influence the sheep. This optimisation considered the minimization of the sheep distance to the goal whilst maintaining flock cohesion.

From the obtained results, it is possible to conclude that the sheep remained cohesive throughout the herding time if and only if they could locate enough of their neighbours at all times. The simulation results showed the ability of the proposed approach to improve the performance of a shepherd in herding up to 100 sheep while considering their limited sensing range. The proposed method was also able to outperform the model proposed by Strombom [11] and CADSHEEP [12].

Future work will investigate our approach in the presence of obstacles and try to account for the topological change of the flock as the shepherd approaches in the identification of herding points. Moreover, the problem addressed can be investigated in multi-objective optimization context, the type of forces influencing the agents like drag force may be considered, and 3-d environment can be used for further investigation of the proposed method performance.

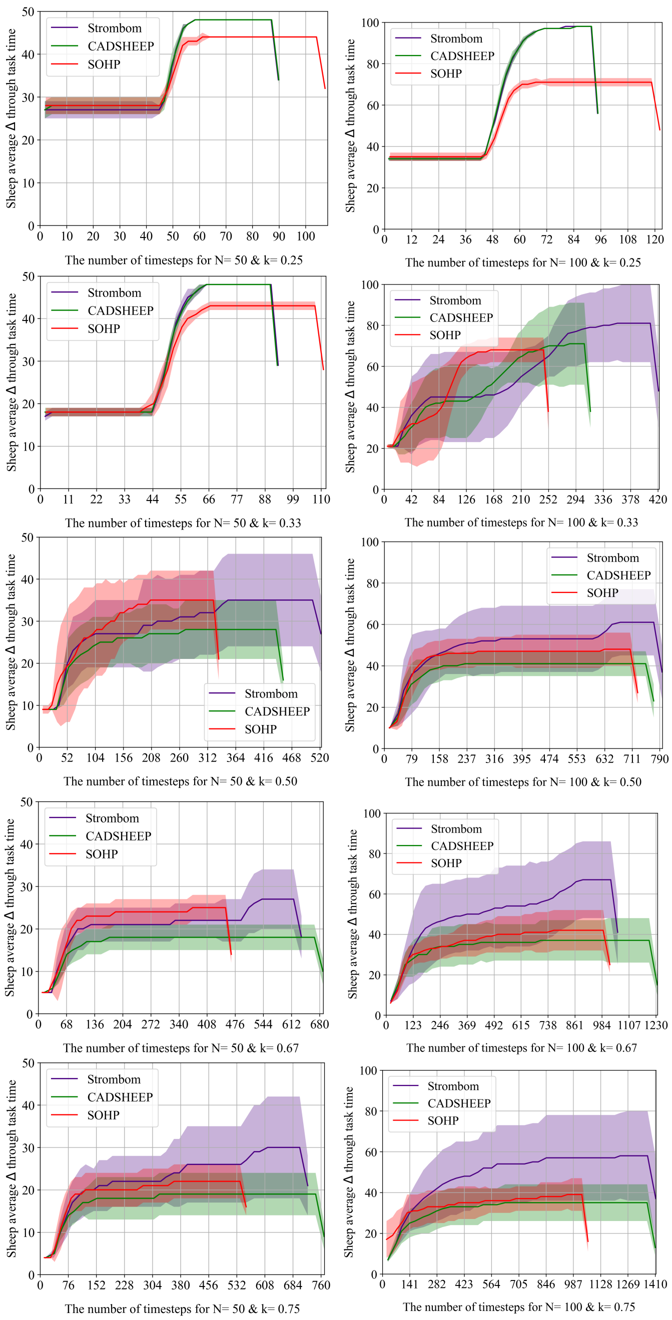

Fig. 2. The change in the average node degree $(+/-)$ one standard deviation for N=50 (left) and N=100 throughout the task time measured at different k=1/4,1/3,1/2,2/3,3/4 from top to bottom

VI. Acknowledgement

This publication was made possible by Grant No N62909-18-1-2140 from Office of Naval Research Global (ONRG). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the ONRG.

References

[1] D. Strömbom and A. J. King, "Robot Collection and Transport of Objects: A Biomimetic Process," *Frontiers in Robotics and AI*, vol. 5, no. May, pp. 1–7, 2018.

[2] 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.

[3] S. Rodriguez and N. M. Amato, "Behavior-based evacuation planning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 350–355, 2010.

[4] J. M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodríguez, and N. M. Amato, "Shepherding Behaviors," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 4, pp. 4159–4164, 2004.

[5] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 109–117, 2000.

[6] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[7] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE, 1995, pp. 1942–1948.

[8] S. Etemadi, R. Vatankhah, A. Alasty, G. R. Vossoughi, and M. Boroushaki, "Leader connectivity management and flocking velocity optimization using the particle swarm optimization method," *Scientia Iranica*, no. 5, pp. 1251–1257.

[9] 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, 2013*, 2015.

[10] A. Procaccini, A. Orlandi, A. Cavagna, I. Giardina, F. Zoratto, D. Santucci, F. Chiarotti, C. K. Hemelrijk, E. Alleva, G. Parisi, and C. Carere, "Propagating waves in starling, Sturnus vulgaris, flocks under predation," *Animal Behaviour*, vol. 82, no. 4, pp. 759–765, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.anbehav.2011.07.006

[11] D. Strombom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. T. Sumpter, and A. J. King, "Solving the shepherding problem: heuristics for herding autonomous, interacting agents," *Journal of The Royal Society Interface*, vol. 11, no. 100, pp. 20 140 719–20 140 719, 2014.

[12] S. Elsayed, H. Singh, E. Debie, A. Perry, B. Campbell, R. Hunjel, and H. Abbass, "Path planning for shepherding a swarm in a cluttered environment using differential evolution," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 2194–2201.

[13] H. Abbass, J. Harvey, and K. Yaxley, "Lifelong Testing of Smart Autonomous Systems by Shepherding a Swarm of Watchdog Artificial Intelligence Agents," 2018. [Online]. Available: http://arxiv.org/abs/1812.08960

[14] H. El-Fiqi, B. Campbell, S. Elsayed, A. Perry, H. K. Singh, R. Hunjet, and H. A. Abbass, "The Limits of Reactive Shepherding Approaches for Swarm Guidance," *IEEE Access*, vol. 8, pp. 214 658–214 671, 2020.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.