

Received September 24, 2020, accepted October 29, 2020, date of publication November 11, 2020, date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037325

The Limits of Reactive Shepherding Approaches for Swarm Guidance

HEBA EL-FIQI¹, (Member, IEEE), BENJAMIN CAMPBELL²,
SABER ELSAYED¹, (Member, IEEE), ANTHONY PERRY²,
HEMANT KUMAR SINGH¹, (Senior Member, IEEE), ROBERT HUNJET²,
AND HUSSEIN A. ABBASS¹, (Fellow, IEEE)

¹School of Engineering and Information Technology, UNSW Canberra, Canberra, ACT 2610, Australia

²Department of Defence, Defence Science and Technology Group, Edinburgh, SA 5111, Australia

Corresponding author: Hussein A. Abbass (h.abbass@unsw.edu.au)

This work was supported by the U.S. Office of Naval Research–Global (ONR-G) Grant.

ABSTRACT Sheepdogs smartly herd a flock of sheep and guide them towards a goal. A single dog can herd a few hundred sheep in easy to navigate environments. Understanding the interaction space between the sheepdogs, sheep and the environment is important due to the possibility of transferring this knowledge to solve practical swarm robotics problems. This interaction space is a complex mesh of influencing factors. We scrutinize this interaction space to identify areas where the complexity of the herding problem changes from low (easy to solve) to high (harder to solve or becoming unsolvable) complexity. In particular, we study reactive models for shepherding, whereby agents respond directly to stimuli in the environments by fusing the set of force vectors influencing their behaviour. We present an enhanced shepherding model with higher success rate than its predecessor. We investigate four key factors that influence the complexity of the problem: the relative speed between the sheepdog and sheep, the spatial configuration of the sheep at the start of the task, the number of sheepdogs, and the density of obstacles in the environment. We discovered a phase transition in shepherding resulting from the interaction between the number of sheepdogs and obstacles. The phase transition occurs as the density of obstacles range from 0.2% for a single shepherding agent to 5% for 10 shepherding agents. During this phase transition, the problem changes from being an easy problem where the flock gets collected quickly, to a hard one where the overall herding task becomes utterly not achievable using reactive approaches.

INDEX TERMS Task complexity, shepherding, swarm robotics, swarm guidance.

I. INTRODUCTION

What makes a problem hard? Why are some problems that appear from the outset to possess similar characteristics, much harder than other similar problems? What is the true source of complexity *in the problem space*? These questions have been the subject of inquiry in artificial life [1] and complexity science [2] for decades. They have gained interest from a wide variety of research areas including optimization [3] and machine learning [4]. More recently, Swarm Robotics [5] have been confronted with the same question; what makes a swarm robotics problem hard?

The literature of swarm robotics has seen many recent attempts to solve the guidance and control problem [see for

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez¹.

example Chung *et al.* [6]]. One approach for guidance and control that started to gain attentions in recent years is *shepherding*. Long *et al.* [7] surveys the shepherding literature and offers a comprehensive coverage of the topic.

In shepherding [8], a human, the shepherd, commands a cognitive actuator, the sheepdog, to exercise a level of influence on a swarm, the sheep, to achieve an intent. Once a command is issued, the cognitive actuator needs to autonomously achieve the intent of the command. For example, if the farmer requests the sheepdog to herd the sheep, the sheepdog may perform a series of two primitive behaviours to achieve the intent [9]. The first behaviour is “collecting”, whereby a stray sheep needs to be brought back to the flock. The second is “driving”, whereby a flock needs to be influenced such that the repulsive response from the sheepdog leads them to the goal area.

The shepherding problem definition is generic enough that it has been transferred to many applications including crowd control [10], rerouting and evacuation [11], keeping birds away from airports [12], collecting oil spills [13], and herding living cells to perform collective migrations [14]. A recent survey on the topic lists more domain of applications [7].

More recently, Strömbom *et al.* [9] adopted a similar model to Miki and Nakamura [15], but with different naming conventions. The guidance behavior was renamed to driving, the flock making behavior renamed to collecting, and the remaining two behaviors maintained their name. Strömbom *et al.* calibrated their model using real-world data and validated that the model replicated the macro behaviors displayed by real sheep and sheepdogs. To the contrary of Lien *et al.*, Strömbom *et al.* were able to scale their model to herd up to 300 sheep with a success rate of 80%.

While shepherding is a fascinating problem on a fundamental level, understanding the complexity (and its causes) of the problem is paramount to identifying guidance and control solutions which are able to scale up in a practical setting. The need for such control solutions is evident when we consider the problem of sky shepherding, where the sheepdog can fly over obstacles [16], [17]. A number of drones act as *aerial sheepdogs* attempting to herd a flock of *sheep* (which may themselves be drones or unmanned systems). It seems plausible to assume that the larger the number of sheep, the more complex the problem is. However, lessons from complexity science have revealed that true complexity arises from the non-linearity inherent in a system and the interactions of systems-of-systems, and not from the size of a particular system [18]. For example, today, we could guarantee to solve linear problems, mostly in a polynomial time, with millions of variables. Still, a highly nonlinear problem with a dozen variables could be truly hard, if not impossible, to guarantee a solution for.

Throughout the literature, there is a great deal of variations in how a shepherding task is judged complete. The shepherding goal can be an object [19], a defined area [20], a pen with a gate [21] or simply a selected corner of the movement zone [9]. Another variation is in the complexity of the environment, i.e. in some shepherding literature there are obstacles in the environment [22], [23], while in others, there are not [9], [24]. This lack of standardisation of complexity is problematic as Linder and Nye [25] showed that variations in the design of the shepherding problem impact the performance of different shepherding strategies and the claims made on one model vis-a-vis another.

A primitive form of the relationship between complexity and successful shepherding was investigated in Lien *et al.* [26] by varying the tendency of the sheep agents to scatter, or what we termed as the collision avoidance force in this paper. The authors found that the performance of shepherding systems, i.e. time steps to completion, decreased as the complexity of the problem (sheep scattering) increased. The rate at which this degradation occurred had an inverse

relationship to the number of shepherding agents that were used i.e. the more shepherds, the better.

Considering the findings from Lien *et al.* [26], it is not surprising that multi shepherd control is a popular approach and is used by many researchers [11], [15], [20], [23], [26]–[28]. In order to select the optimal position for each shepherding agent in a multi shepherd system, the relative locations of the other shepherds as well as the sheep must be considered. Although it has been demonstrated that multi shepherd co-operation could naturally emerge [11], [29], a more common approach is to select a formation and assign each shepherding agent a position within the formation [23], [26]–[28], [30].

Researchers [23], [26], [27] used an arc shaped formation for driving the flock, and others [23], [26] utilise a fixed formation where a shepherd position is strictly assigned based on the size of the flock and the relative position of the flock to the goal. The focus of Masehian and Royan [23] was on selecting the optimal time for shepherding agents to leave the driving formation in order to collect sheep that are separated from the flock (scattered). In contrast, Kalantar and Zimmer [27] proposed “Deformable formations” with the ability to respond to the shape of the flock. They suggest that a deformable formation is a more efficient shepherding strategy and could reduce the frequency of scattering.

Another formation that has been used is caging, where the shepherds surround the flock, then coordinate their movements towards the goal. Bat-Erdene and Mandakh [28] utilized a square shaped cage formation. Interestingly, in their proposed solution the shepherding agents are heterogeneous with two distinct roles, “corner robot” and “sideline robot”. This strategy was necessary due to their reliance on physical barriers rather than repulsive forces to contain the sheep, necessitating specialized hard-wired behaviors depending on position in the flock.

The aim of this paper is to focus on the interaction space between the sheepdog, the sheep, and the environment to understand a few sources of complexity for shepherding. We consider an instance of a problem to be more complex than another if, assuming everything other the complexity factor being investigated are constant, the former has a lower success rate and/or require a higher completion time than the latter.

We carry out the investigation in two stages. In the first stage, we limit the number of sheepdogs to one and study the relationship between the density of sheep and the relative speed of the sheepdog to the individual sheep on the one hand, and task success indicators (success rate and completion time), on the other. We assume an obstacle free environment. This experiment allows us to identify the phase transition where problem difficulties move from a problem that is simple to solve, to a problem that can't be solved. To put it simply, the experiment helps us to answer the question: how many sheep can we collect with a single sheepdog at different sheepdog's speed profiles.

In the second stage, the interactions between the number of sheepdogs and the density of obstacles in the environment are examined. It logically follows that an increase in the number of sky shepherds for the same flock size will likely have a positive effect on addressing task's complexity. Therefore, we study the interaction between the number of sheepdogs and the density of obstacles in the environment, and the impact of this interaction on task success indicators.

The contributions of this paper are:

- We present a modified model for shepherding that avoids disturbing the flock when the sheepdog approaches. This is achieved by assigning a path to the sheepdog which maintains a distance to the flock greater than its influence range.
- We present a systematic study to investigate factors for the complexity of the shepherding problem.
- The results identify factors which contribute to the difficulty of the shepherding problem. These assist us in designing difficult problem instances for testing swarm guidance algorithms.

The rest of this paper is organised as follows: The single and multiple sheepdog shepherding models, including our modification to Strömbom, are first discussed. We then present the mathematical metrics for task complexity followed by experimental design. Analysis then conclusions are then drawn.

II. SINGLE AND MULTIPLE SHEEPDOG SHEPHERDING MODELS

The shepherding model that we are using in this study shares some characteristics with Strömbom's model [9]. The main similarity between the two models is in terms of employing attraction and repulsion forces to model both sheep and sheepdog behaviors. However, the forces are calculated differently and used under different conditions. We will first present the notational system used throughout the manuscript, then present the single-sheepdog model.

A. GENERALISED NOTATIONAL SYSTEM AND ASSUMPTIONS

We denote the set of sheep agents with $\Pi = \{\pi_1, \dots, \pi_i, \dots, \pi_N\}$, where the letters Π and π are chosen as the first character of the Greek word for sheep $\Pi\rho\rho\beta\alpha\tau\omicron$ and denote the set of shepherd agents with $B = \{\beta_1, \dots, \beta_j, \dots, \beta_M\}$, where the letters B and β are chosen as the first character of the Greek word for Shepherds $B\omicron\sigma\kappa\rho\acute{o}\varsigma$.

We denote the set of behaviors in the simulation with $\Sigma = \{\sigma_1, \dots, \sigma_K\}$, where the letters Σ and σ are chosen as the first character of the Greek word for behaviour $\sigma\nu\mu\pi\epsilon\rho\iota\phi\omicron\rho\rho\acute{\alpha}$.

In the model, agents are initialized in a squared area. We use u to denote the unit, where u could be described in meters or other convenient units of length.

All the agents are assumed to have access to a shared centralized information database. The database is managed by a centralized task manager agent, which has access to

TABLE 1. Generalised shepherding notational system.

Symbol	Description
$\Lambda_{\pi_i}^t$	Local centre of mass for π_i at time t
$\Lambda_{\beta_j}^t$	Local centre of mass for β_j at time t
$\Gamma_{\pi_i}^t$	Global centre of mass for π_i at time t
$\Gamma_{\beta_j}^t$	Global centre of mass for β_j at time t
$P_{\beta_j}^t$	Position of agent β_j at time t
$P_{\pi_i}^t$	Position of agent π_i at time t
$P_{\beta_j\sigma_1}^t$	Position for driving by agent β_j at time t ; $R_{\pi\pi}\sqrt{N}u$ behind the flock
$P_{\beta_j\sigma_2}^t$	Position for collection by agent β_j at time t ; $R_{\pi\pi}u$ behind the furthest agent
$F_{\pi_i\beta_j}^t$	Force vector, repulsion of π_i agent away from β_j agents at time t
$F_{\pi_i\pi_{-i}}^t$	Force vector, repulsion of π_i agent away from other $\pi_{k\neq i}$ agent at time t
$F_{\pi_i\Lambda_{\pi_i}^t}^t$	Force vector, attraction to Local Centre of Mass for the neighbours of a π_i agent at time t
$F_{\pi_i\epsilon}^t$	Force vector, jittering movements by the π_i agent at time t
$F_{\pi_i}^t$	Force vector, movement vector of the π_i agent at time t
$F_{\beta_j cd}^t$	Force vector, driving and collection vector of the β_j agent at time t
$F_{\beta_j\epsilon}^t$	Force vector, jittering movements by the β_j agent at time t
$F_{\beta_j}^t$	Force vector, movement vector of the β_j agent at time t
$\Omega_{\pi_i\pi}$	Set of π agents (neighbourhood) a π_i agent operates on
$\Omega_{\beta_j\pi}$	Set of π agents a β_j agent operates on
Ω^t	Set of π agents at time t that have not reached the goal; that is, the remaining agents to complete the mission.
$\Omega_{\beta_j\pi}^t$	Set of π agents a β_j agent operates on at time t
$ \Omega_{\pi_i\pi} $ (1-200)	Number of π agents (neighbourhood) a π_i agent operates on
$ \Omega_{\beta_j\pi} $ (20)	Number of π agents a β_j agent operates on
$ \Omega^t $	Number of π agents at time t that have not reached the goal; that is, the remaining agents to complete the mission.
$ \Omega_{\beta_j\pi}^t $	Number of π agents a β_j agent operates on at time t
$S_{\pi} (1 u/t^2)$	Maximum speed of π per time step
$S_{\beta} (1.5 u/t^2)$	Maximum speed of β per time step
S_{β}^t	Speed of β at time t
\mathbb{D}	The minimum distance required between any sheep and the goal to announce successful shepherding and task completion
η (0.05)	Probability of moving per time step while grazing
$\theta_{\beta} (\pi/2)$	Blind angle for β

all sheep locations, sheepdogs' locations, and the goal. This removes complexity resultant from related delays (e.g. multiple communications interactions between entities) which are out of scope for this study.

At each time-step, the task manager assigns the collecting tasks first; then if any remaining sheepdogs were not assigned a collection task, the task manager assigns them a driving task.

The above assumptions eliminate confounding factors that could result from a decentralized setting, such as factors associated with local communications among the agents, or loss of situation awareness due to distributed sensing and/or decision making. The authors acknowledge that such problems (distributed decision making, consensus, communication constraints and algorithmic complexity) are important for the application to real robotic systems, and highlight these areas as future work.

B. PROPOSED SINGLE-SHEPHERD MODEL

First, we will describe our model. Then, we will discuss similarities and differences from our model and the one used by [9].

TABLE 2. Parameters and their assumed value(s) in both experiments in this manuscript.

Symbol	Description	Value
L	Length and Width of Environment	50
N	Cardinality of Π	100
M	Cardinality of B	1 ... 10
$R_{\pi\beta}$	π sensing range for β	65
$R_{\pi\pi}^s$	π sensing range for π	3
$R_{\pi\pi}^a$	π avoidance range for π	0.4
$R_{\pi o}^a$	obstacle avoidance range for π	5
$R_{\beta\beta}$	β avoidance range for β	2
$W_{\pi\pi}$	π repulsion strength from π	2
$W_{\pi\beta}$	π repulsion strength from β	1
$W_{\pi\Lambda}$	π attraction strength to Λ	1.05
$W_{\pi o}$	π repulsion strength from obstacles	3
$W_{\pi v}$	Strength of π previous direction	0.5
$W_{e\pi}$	Strength of sheep π angular noise	0.3
$W_{e\beta}$	Strength of shepherd β angular noise	0.3
$W_{\beta\beta}$	β repulsion strength from β	0.5
S_β	Maximum speed of β per time step	2
R2	Minimum separation distance between a sheepdog and a sheep	4
R3	length of the influencing distance for the sheepdog	10
P_G^t	Position of goal at time t , classically remains unchanged	(25, 50)
	Goal Radius	10
I_{II}	Sheep Initialisation Boundary	(20, 30, 20, 30)
	Sheep Initialisation Pattern	P1-P6
I_B	Sheepdog Initialisation Boundary	(15, 35, 45, 50)
	maximum Number of Steps	2000

A sheep will be affected by the following behaviors:

- 1) The sheep will be repelled from any sheepdog that is approaching;
- 2) A sheep will be attracted to the local center of mass of its flock (cohesion);
- 3) A sheep will try to avoid colliding with other sheep;
- 4) A small jittering behaviour while grazing;
- 5) The sheep will also be affected by its previous direction; and
- 6) The sixth behaviour that we add to this model is the obstacle avoidance behaviour, where the sheep repulse away from obstacles with a stronger force than the force they use to repulse away from each others to avoid collision.

A sheep will try to escape from any sheepdog that exists within a sheep to sheepdog detection range $R\pi\beta$. The repulsive force is regulated using function M_1 presented in Equation 1, which regulates the collision avoidance forces between any two objects according to the distance and speed ratio of the two objects. In the case of the sheep trying to escape the sheepdog, s is substituted with the ratio between the sheepdog speed and the sheep speed, d is the distance between the sheepdog and the sheep, r_{max} is the maximum distance that a sheep can sense a sheepdog, and r_{min} is the minimum separation distance that the sheepdog has to maintain to any sheep.

$$M_1(s, d, r_{min}, r_{max}) = s \times e^{-2d/(r_{max}-r_{min})} \quad (1)$$

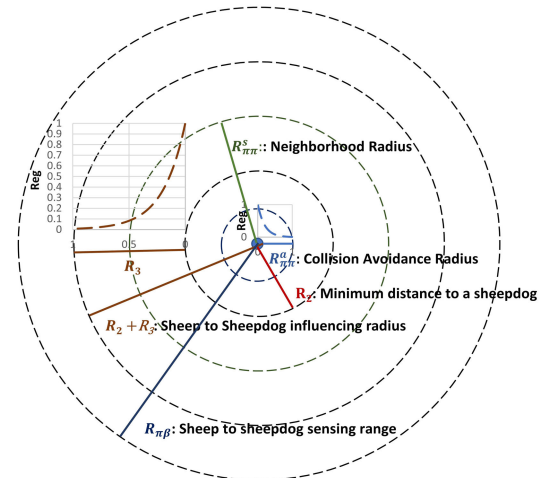


FIGURE 1. Sheep influencing circles.

The cohesion force, $F_{\pi_i\Lambda_{\pi_j}}^t$, toward the local center of mass, is calculated based on the sheep that are sensed in the neighborhood. Any sheep detected within a sensing range R_{π}^s is considered as a neighboring sheep, and its location will be part of the local center of mass calculated by that sheep.

The collision avoidance force $F_{\pi_i\pi_j}^t$ is a repulsion force that is calculated as the cumulative sum of the regulated forces away from any sheep within R_{π}^a distance from this sheep. The regulation of those individual forces is calculated based on the distance between the two sheep. The regulation function M_2 is defined as $\frac{1}{\sqrt{(N)}}M_1$. The jittering force $F_{\pi_i\epsilon}^t$ is a small random force across the two dimensions that will affect the sheep location. It models the grazing when no other forces are affecting the sheep direction. As the sheep changes its direction, the previous heading direction will have some force $F_{\pi_i}^{t-1}$ impacting the sheep direction.

If the sheep detects an obstacle within $R^a\pi o$, this sheep will be repelled from the obstacle in the same way that it avoids sheepdogs, except that in the regulation function in Equation 1, the s parameter is set to the sheep agent speed due to the static nature of the obstacles, and the maximum distance r_{max} is defined by the obstacle detection range $R^a\pi o$. The total forces that affect the sheep's next location is a cumulative sum of the weighted forces of the individual forces discussed earlier as in Equation 2.

$$F_{\pi_i}^t = W_{\pi v}F_{\pi_i}^{t-1} + W_{\pi\Lambda}F_{\pi_i\Lambda_{\pi_j}}^t + W_{\pi\beta}F_{\pi_i\beta_j}^t + W_{\pi\pi}F_{\pi_i\pi_j}^t + W_{e\pi_i}F_{\pi_i\epsilon}^t + W_{\pi o}F_{\pi_i o}^t \quad (2)$$

A sheepdog will be either collecting, driving, or approaching its driving/collecting locations under a general mission of herding the sheep toward a goal location. However, the driving and collecting points should be selected on the perimeter of the middle circle shown in Figure 2. If the sheep are clustered, the sheepdog will identify the driving location at a distance $R1 + R2$ behind its local center of mass $\Lambda_{\beta_j}^t$

and pointing toward the goal. $R1$ is the maximum allowed distance that a sheep can be located further away from the sheepdog local center of mass and continues to be assumed within the cluster. If the sheep goes beyond $R1$, then it needs to be collected and returned to the group. If more than one sheep were outside the area of the flock, then, the sheepdog will identify the furthest sheep from the sheepdog's local center of mass, and set a collecting point behind that sheep toward the $\Lambda_{\beta_j}^t$ to herd the sheep back to the group. $R2$ is the minimum separation distance the sheepdog needs to maintain in order not to disturb the sheep. Therefore, if the sheepdog is driving, it maintains $R1 + R2$ distance from $\Lambda_{\beta_j}^t$, which indicates minimum separation from the furthest sheep of the clustered flock. If the sheepdog is collecting, it maintains $R2$ separation distance from the most dispersed sheep from the flock, which will induce a maximum herding force toward that sheep. A small jittering force is applied to the sheepdog to avoid deadlock situations.

The force toward either the collecting point or driving point is identified as $F_{\beta_j cd}^t$, and is calculated as the normalized vector from the current sheepdog location toward that point. The total force that affects the sheepdog at any time step is calculated using Equation 3.

$$F_{\beta_j}^t = F_{\beta_j cd}^t + W_{e\beta_j} F_{\beta_j \epsilon}^t \quad (3)$$

We consider the sheepdog as a repulsive agent, no planning/cognitive capabilities are assumed. However, a method of circular path approach is enacted to minimise unnecessary dispersion of the flock. To enable this, we constrain $F_{\beta_j}^t$'s application to the sheepdog in the following manner:

- The sheepdog should approach its driving/collecting points using a circle around the local center of mass, which has a radius equals to the sum of $R1$, $R2$, and $R3$. This sum represents the distance that ensures the presence of the dog, as it approaches the collecting or driving points, has no impact on the clustered sheep.
- The sheepdog is only allowed to enter that circle if it is within a minimal distance of its target collecting/driving point.
- If the sheepdog next position is trying to breach this circle without being close to the target collection/driving point, a correcting equation to the location is applied to keep it at the boundary of the circle, with one step closer to its target location.

fn defines the radius of the minimum circle that can be used to accommodate all sheep. The sheep are considered clustered if they are located within fn distance from the sheepdog global center of mass. To calculate fn , we start by defining a circle around each sheep with a radius of $R_{\pi\pi}^a$, representing the collision avoidance range between two sheep. A square that holds this circle will be of side length $2R_{\pi\pi}^a$, and has an area of $(2R_{\pi\pi}^a)^2 = 4(R_{\pi\pi}^a)^2$.

To allow the sheep some freedom to spread, we reserve an area of $4(R_{\pi\pi}^a)^2$ for a single sheep, whereby N sheep occupy an area of $4N(R_{\pi\pi}^a)^2$. The side length of this square

is $2R_{\pi\pi}^a \sqrt{N}$, with a diagonal $\sqrt{2}$ of its side.

$$fn = R_{\pi\pi}^a \sqrt{2N} \quad (4)$$

Therefore, fn is calculated using Equation 4, which provides a smaller circle than $fn = R_{\pi\pi}^a N^{\frac{1}{3}}$ suggested in [9]. For example, for $N=100$, and $R_{\pi\pi}^a$, fn using Equation 4 suggests a radius of 28.28 compared to 43.09 used in Strömbom's fn equation. The new equation for fn has been validated to provide enough radius to accommodate N circles of radius $R_{\pi\pi}^a$ using data collected for the optimization problem of packing equal circles in a circle for N values up to 2000 agent.¹

The $R2$ parameter could be fixed to represent the minimum separation distance between a sheepdog and a sheep. The sheep to sheepdog escaping force is regulated using $M1(s, d, R2, R2 + R3)$ Equation 1. At $R2$ distance, the sheep receives the highest repulsion force. At a distance of $R2 + R3$, the regulated repulsion force is very low.

During collecting behavior, the driving position is at distance $R2$ of the sheep, and will receive the maximum force. However, during the driving behavior, the furthest sheep from the global center of mass can be located at a distance to the sheepdog greater than $R2 + R3$, for example, when it is at the other side of the flock. To ensure that the sheepdog will at least influence one sheep while driving, $R2 + R3$ must be greater than $R1 + R2$; thus, $R3 \gg fn$.

C. SUMMARY OF THE SIMILARITIES AND DIFFERENCES TO Strömbom et al.'S MODEL

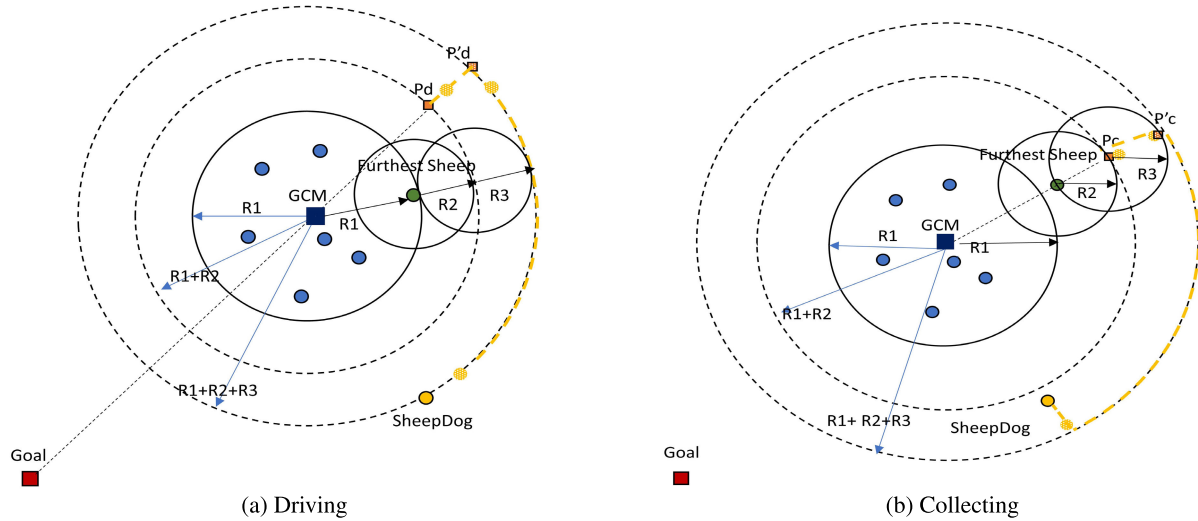
Our proposed model's paramterisation was guided with the parameters resultant from the calibration done by Strömbom et al. The similarities between our model and Strömbom's model can be summarized in the following points:

- 1) Both models use attraction and repulsion forces to model shepherding behaviour.
- 2) Both models consider the previous direction effect.
- 3) Attraction to neighbors, collision avoidance, and escaping from sheepdogs are modelled in the same manner in both models.
- 4) Both sheep and sheepdogs have jittering behaviour.
- 5) The two models share the same weights for the forces shared between the two models.

The differences between our model and Strömbom's model can be summarized in the following points:

- 1) The sheepdog approaches the driving and collecting locations via a circular path in our model rather than a straight line in Strömbom's.
- 2) In Strömbom's model, if the sheep detect no sheepdog, it maintains its previous location, no attraction to other sheep or collision avoidance forces are applied. In our model, the sheep will always be affected by all other forces, the exception being force to escape (repulsion from) a sheepdog, which in the absence of an influencing sheepdog is nil.

¹<http://www.packomania.com/>


FIGURE 2. Sheepdog operating circles.

- 3) In our model, the sheepdog operates using three circles: the first defines an imaginary boundary of radius $R1$ around the sheep LCM. The second defines a banned area that the sheepdog cannot enter to keep the sheep grouped together. The radius of this circle is $R1+R2$, where $R2$ is the minimum separation distance between the sheepdog and any sheep. The third circle defines the boundary of the area where the sheepdog can operate beyond it without influencing any sheep. The sheepdog uses this third circle boundary to reach its driving and/or collecting points without influencing any sheep. The radius of this circle is $R1+R2+R3$, where $R2+R3$ defines a distance, where the sheepdog has no influence on the sheep beyond it.
- 4) The function, $F(N) = R_{\pi\pi}^a N^{\frac{2}{3}}$ —used in Strömbom’s to estimate if the sheep are grouped—was updated to become a part of $R1$ ’s definition in our model. $R1 = R_{\pi\pi}^a \sqrt{2N}$ provides a smaller radius than $F(N)$, and will yet accommodate N number of sheep within a circle. In our model, $R1$ affects both driving and collecting behaviors.
- 5) Driving and collecting positional equations were updated to utilize the circles model.
- 6) In our model, sheep definition of the neighborhood is based on sensing range, not on the closest n number of sheep as in Strömbom’s model. This modification is designed to reflect a robot’s limited sensing capabilities.
- 7) In our model, two types of force regulations with collision avoidance and escaping behaviors are used to reflect the effect of the distance on the action. The closer the sheepdog, the higher the escaping force.

D. PROPOSED MULTI-SheepDog SHEPHERDING

To evaluate the complexity of the multi-sheepdog shepherding problem, several decisions with respect to division of

labour and communication need to be made. We will start first with the assumptions we made to reduce the number of confounding factors that could impact our complexity analysis.

1) COLLECTING

The task manager scans the locations for all the sheep, then calculates its global center of mass Γ^t to identify whether or not the sheep are grouped in a single cluster or not using a radius $\Omega_{\neq\Gamma}^t$.

$$\Omega_{\neq\Gamma}^t = \forall \pi_i \in \Pi, \|\Gamma^t - P_{\pi_i}^t\| > f(N) \quad (5)$$

If $\Omega_{\neq\Gamma}^t \neq \emptyset$, then the task manager starts with the first sheepdog β_j , and finds the closest sheep c in the out-of-flock set $\Omega_{\neq\Gamma}^t$ to the sheepdog. It then allocates the task of collecting the sheep identified in the previous step to this sheepdog.

$$\pi_k = \min \|P_{\beta_j}^t - P_{\pi_i}^t\| : \pi_i \in \Omega_{\neq\Gamma}^t \quad (6)$$

$$\text{roster}(\beta_j) = \text{Collect}(\text{sheepID} : \pi_k, \text{GCM} = \Gamma^t) \quad (7)$$

This allocation task is stored in a roster accessible by all sheepdogs. Each row in this roster will have the sheepdog ID, and the assigned task (either collecting or driving). When a sheepdog looks up its ID in the roster table, and finds that it is being assigned a collecting task, it looks for the relevant collecting information in a shared collecting knowledge table. In that table, the sheepdog will find the assigned sheep ID, location, and GCM provided by the task manager. It then derives its collecting position behind the sheep and toward the GCM. Sheep π_k gets removed from $\Omega_{\neq\Gamma}^t$, and the task manager repeats the same steps for the next sheepdog until either $\Omega_{\neq\Gamma}^t = \emptyset$, or all sheepdogs in the flock have been assigned a collecting task.

2) DRIVING

The task manager starts from the next available sheepdog after the collecting assignments have been assigned. If all

sheepdogs are allocated a collecting task, the driving task does not start. The sheepdogs assigned during driving will make a formation using the following coding system. The sheepdog in the center of the formation will be given the code C . The one on its left will be called CL . The one on the left of CL will be called CLL and so forth. The one on the right of C is called CR . The coding continues with CRR , $CRRR$ and so forth.

The first available sheepdog in the list for driving becomes the central sheepdog, C . The driving point for C is chosen to be behind Γ^t , and facing the goal location. The distance is defined based on the shepherding model selection.

$$\text{roster}(\beta_j) = \text{Drive}(\text{Role} : 'C', P = P_{\beta_j}^t, GCM = \Gamma^t) \quad (8)$$

Then, the next available sheepdog is assigned the task to drive around a surface of a circle surrounding the Γ^t towards the CL position relevant to C . The distance between CL and C is a hyper-parameter. We select to use a driving arc of length $\frac{1}{2}\pi$. Therefore, the angle that is used to calculate the location of CL is calculated using $\frac{1}{2M_1}\pi$, where M_1 is the number of driving sheepdogs remaining in the list (excludes all those assigned to collecting tasks). The next sheepdog is assigned the task of driving at CR on the right of C , and the allocation process continues in the same way.

$$\text{roster}(\beta_l) = \text{Drive}(\text{Role} : 'C', P = P_{\beta_l}^t, GCM = \Gamma^t, \text{RefDog} : 'C', \text{Direction}_{\text{RefDog}} = 'R') \quad (9)$$

As every sheepdog has access to the roster, each sheepdog can lookup its reference dog, and access its location as well. Using information from Equation 9, a sheepdog β_l identifies the angle between its reference sheepdog and the global center of mass using Equation 10, and its distance as well using 11. It then calculates its own angle relevant to the global center of mass using Equation 13 using the output from Equations 11 and 12. Then, it can estimate its driving location using Equation 14

$$\theta_{\beta_l \Gamma^t} = \text{atan}(P_{\beta_l}^t - \Gamma^t) \quad (10)$$

$$R'_{\beta_l \Gamma^t} = \|P_{\beta_l}^t - \Gamma^t\| \quad (11)$$

$$\theta_{\text{dist}} = \frac{1}{2M_1}\pi \quad (12)$$

$$\theta_{\beta_l \Gamma^t} = \begin{cases} \theta_{\beta_l \Gamma^t} - \theta_{\text{dist}}, & \text{Direction}_{\text{RefDog}} = 'L' \\ \theta_{\beta_l \Gamma^t} + \theta_{\text{dist}}, & \text{Direction}_{\text{RefDog}} = 'R' \end{cases} \quad (13)$$

$$P_{\beta_l \sigma_d}^{t+1} = R'_{\beta_l \Gamma^t} \cos(\theta_{\beta_l \Gamma^t}) + \Gamma^t.x, R'_{\beta_l \Gamma^t} \sin(\theta_{\beta_l \Gamma^t}) + \Gamma^t.y \quad (14)$$

Using the design assumptions described above, sheepdogs of lower IDs will have a higher chance to be allocated to collecting tasks than driving ones. Those used for driving, will have a higher chance to be at the center of the formation than at either ends of the formation arc. Other strategies could be adopted including shuffling of IDs, reordering of the sheepdog list based on internal states such as energy/battery level, and/or external states such as locations in the environment.

III. MATHEMATICAL METRICS OF TASK COMPLEXITY

Wood [2] identified three dimensions of task complexity. The first two are static, i.e. they are invariant during the execution of a task. These are component complexity, representing the number of different information cues and acts that are essential for the completion of the task, and coordinative complexity, representing the relationships that exist between the inputs to a task and the products. The third complexity dimension is dynamic complexity, and measures the stability of the relationships between products and task inputs.

In this paper, we focus on the latter dimension of complexity. The task inputs are the independent influencing factors that we manipulate to influence the complexity of a task. The products are the actions performed by a sheepdog. Stability is measured using two measures of performance.

We follow a similar philosophy to *dynamic complexity* [2] and define task complexity as the change in a set of measures of performance associated with a task when influencing factors that impact these measures change.

This definition suggests that complexity is in the *eye of the beholder*, because it is the beholder who defines measures of performance. The influencing factors could be internal to the artificial agent/sheepdog or external. Internal ones reflect the agent's cognitive, behavioral and physical abilities such as the speed of a sheepdog. External ones are related to factors outside the agent's control such as the initial spatial distribution of sheep in the environment and the density of sheep in the environment.

We use the time-to-herd as a measure of performance for shepherding. The shorter the time it takes the sheepdog to collect and drive all sheep to the goal location, the better. Equation 15 represents this relationship, whereby T is the total herding time, T_{σ_i} is the total time the sheepdog spent while adopting behavior σ_i , and T_{natural} is the time spent by the sheepdog between behavior switching. The total time is therefore calculated as:

$$T = \sum_{i=1}^k \sigma_i + T_{\text{natural}} \quad (15)$$

The complexity of shepherding is measured as the rate of change in T due to a change in a non-trivial influencing factor of the sheep and the environment. A factor is non-trivial when it causes a nonlinear change in T . We consider the initial distance between the sheep center of mass and the goal location as a trivial factor due to the natural increase in T due to a simple increase in this distance, which when everything else is constant, will generate a linearly proportional increase in T .

The first two factors investigated in this paper to impact the complexity of the shepherding task are: the speed differential and spatial distribution. The ratio between the speed of the dog and the speed of the sheep is expected to impact the completion time and success of herding. If the sheep is faster than the dog, it is unlikely that the dog will be able to collect the sheep unless the environment is so constrained and the

sheepdog has the ability to plan how to ambush the sheep and drive them to the goal. This case is not considered due to our reliance on reactive agents.

This speed differential in the sheepdog-sheep relationship impacts task performance. The influence of the sheepdog on sheep $M1$ is multiplied by the speed differential $\frac{S_\beta}{S_\pi}$; thus, the magnitude of the influence vector the sheepdog exerts on the sheep increases as the maximum allowed speed for the dog increases.

The second factor is the spatial distribution of the sheep at the start of the task/simulation, which is expected to influence the complexity of completing the shepherding task successfully. We use the Index of Dispersion (I) as a metric for spatial distribution [31]. Describing this concept, consider n circular quadrats are spread over an area according to a uniform distribution. Assume each quadrat, i , has m_i plants, where $m = \{m_i\}_i$. If the plants were allocated to the quadrats entirely at random, m_i would be a random variable from a Poisson distribution, where the mean $E(m)$ equals the variance $Var(m)$; that is, $E(m) = Var(m)$. When this pattern deviates from a random pattern, $Var(m) > E(m)$. The Index of Dispersion is calculated as

$$I = \frac{Var(m)}{E(m)} \quad (16)$$

The Index of Dispersion evaluates the deviation of the spatial distribution of objects according to some biased pattern from the spatial distribution of these objects when they get allocated in the environment completely at random. To randomize the sheep in patterns, we use a method inspired by [32], whereby the area the sheep will be initialized is divided into squares, where λ represents the density of squares (the number of sheep that gets initialized in that square), and μ is the sparsity of the squares.

In the second experiment, we study two influencing factors: the number of sheepdogs and the density of obstacles in the environment. We use the same spatial distribution patterns of sheep used in the first experiment.

In the remaining sections, we will present the design for each experiment, results and discussion.

IV. EXPERIMENTAL DESIGN

A. SINGLE SHEEPDOG EXPERIMENTAL DESIGN

The number of sheep is fixed in these experiments to 100. The sheepdog-sheep speed ratio is varied between 3 to 1 in a step of 0.05. The sheep are initialized in a $10m \times 10m$ area divided into 25 $2m \times 2m$ squares. The sheep get initialized using six different patterns.

The first pattern allows the sheep to be randomly assigned to any square in the initialization region. The second pattern clusters the sheep in a highly dense area. The third initializes the sheep in a linear formation with gaps between clusters that exceed their local sensing range. The fourth initializes the sheep in a v-shape with the vertex of the 'v' pointing towards the opposite direction of the goal, allowing it to align with the driving point. The fifth is similar to the fourth except that

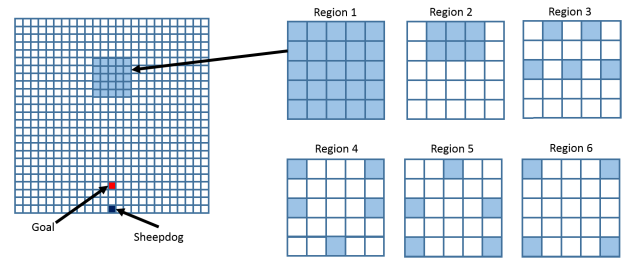


FIGURE 3. The six initialization regions with different spatial distribution.

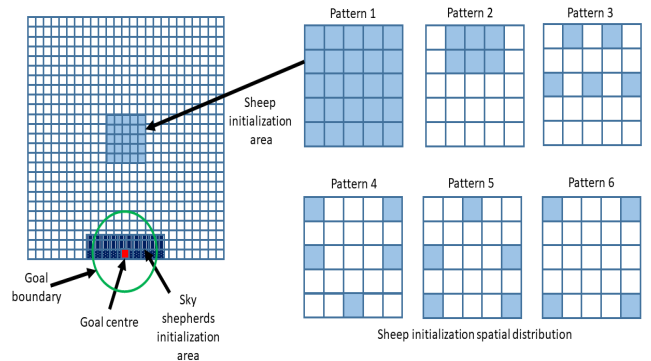


FIGURE 4. Environment initialization.

the vertex of the 'v' is pointing towards the goal, which could cause the driving point to split the sheep. The sixth initializes the sheep at the four corners of the initialization region. The six initialization patterns are shown in Figure 3.

The environment's size is $50m \times 50m$. The goal and initial sheepdog locations are fixed as shown in Figure 3. The sheepdog starts behind the goal, in the opposite direction of the region where the sheep gets initialized, to avoid any initial impact from the dog on the sheep at the initialization stage.

The sheep radius for cohesion is $3m$. This ensures that all sheep initialized in the same $2m \times 2m$ cells are connected and that any sheep at the edge is not connected to a sheep in other cells when the neighboring cells are empty. The maximum cell density in this environment occurs with initialization region P6, where there are four cells available and on average 25 sheep to be initialized in each cell. We select the collision radius to be $40cm$ to ensure sufficient spread in the area.

We set $P_G = \langle 25, 50 \rangle$; $P_\beta^0 = \langle 25, 47 \rangle$; $S_\pi = 1$; the Dog-To-Sheep-Speed-Ratio $\in \{1 + 0.05k \mid k \in [0, 40]\}$; $S_\beta = \text{Dog-To-Sheep-Speed-Ratio} \times S_\pi$; and the initialization-Region-Boundary = $\{xStart, yStart, 10, 10\}$. Each experiment was repeated 30 times with different seeds.

B. MULTIPLE SHEEPDOG EXPERIMENTAL DESIGN

Sheep and sheepdogs are repulsed by obstacles (i.e. there is an interaction with the obstacle), and as such, an increase in obstacle density within the environment represents an increase in the task complexity. We assume a sky shepherding based approach, that is a drone or similar system is being used to herd the flock as per [17], [33]. Therefore,

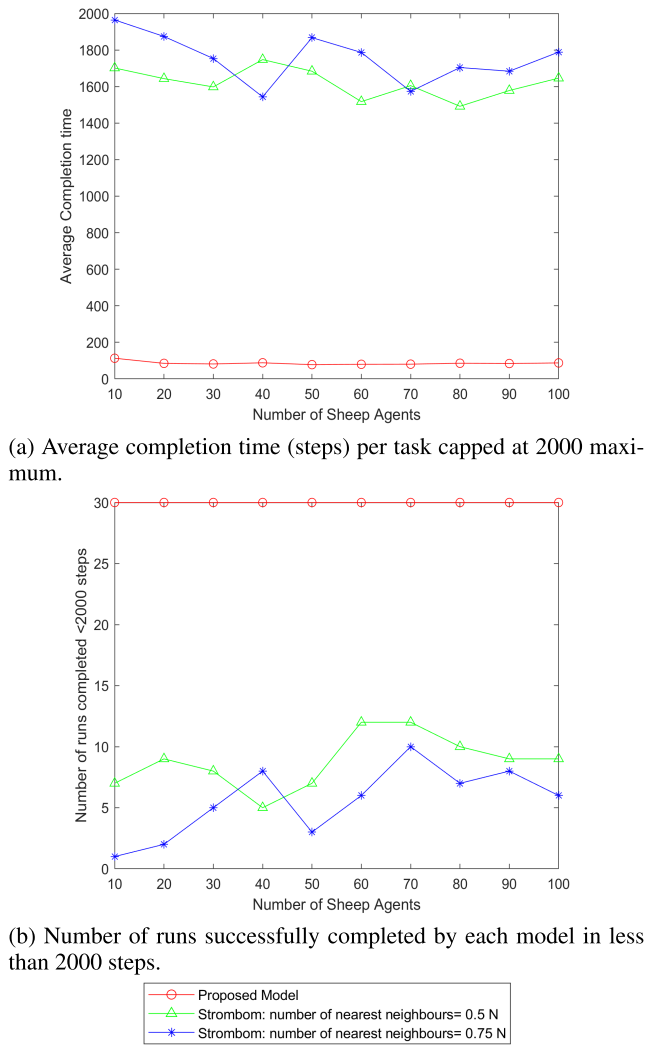
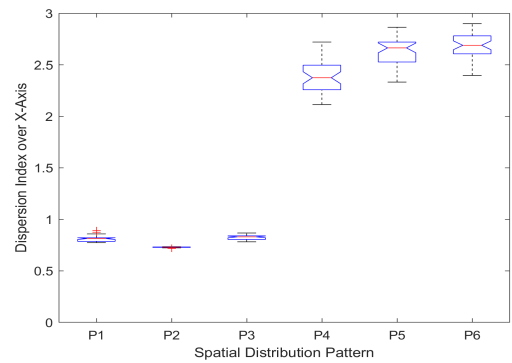


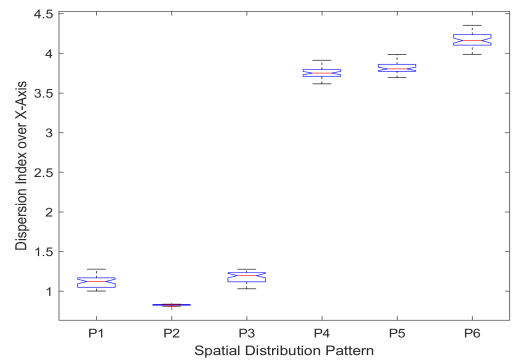
FIGURE 5. Comparison between Strömbom's model and the proposed model.

the sheep are affected by obstacles, however, the sky shepherd is not. The environment is of size 50×50 meters, contains 100 sheep, and one to ten sky shepherds. Sheep move at speed 1 m/s, and sheepdog at a maximum speed 1.5 m/s; thus, the sheepdog-to-sheep speed ratio is 1.5:1. In this environment we instantiate a circular goal area of radius of 10 meters centered midway along the south boundary of the field. The sheepdog initialization area is centered 5 meters north of the goal spaced at 20×5 meters. Sheep are initialized in a 10×10 meters area located in the middle of the field, with the sheep dispersed in 6 different patterns as per figure 4.

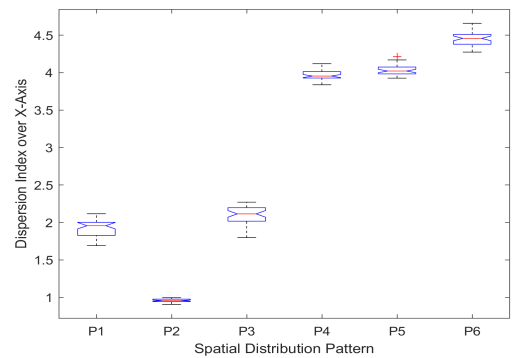
As before, the sheep initialization area is split into 25 squares of 2×2 meters numbered 1 to 25, left to right, top down numbering. The same patterns used in the first experiment for sheep initialization are maintained. Obstacles have a radius of 1 unit and are spread uniformly in the environment with a systematically varied density from 0 to 10% in increments of 0.2%.



(a) Dispersion index over X-axis for all sheep.



(b) Dispersion index over X-axis for the top 25% dispersed sheep.



(c) Dispersion index over X-axis for the top 5% dispersed sheep.

FIGURE 6. The dispersion index distribution in different environments.

Clearly the existence of obstacles is expected to increase the number of steps it will take to herd the sheep due to the need to navigate through longer routes and the difficulties to maintain the flock when the sheep go around obstacles due to the repulsive collision avoidance force and the cascading effect this has on the flock. A maximum simulation duration of 8000 time-steps was deemed appropriate via empirical analysis. The simulation is repeated 30 times for each configuration with different seeds. The parameters used in the simulation are summarized in Table 2.

V. EXPERIMENTAL ANALYSIS

A. COMPARING PROPOSED AND Strömbom's MODELS

An evaluation for the performance of Strömbom's model compared to the enhancements proposed in the new model was performed. We compared the completion time of the

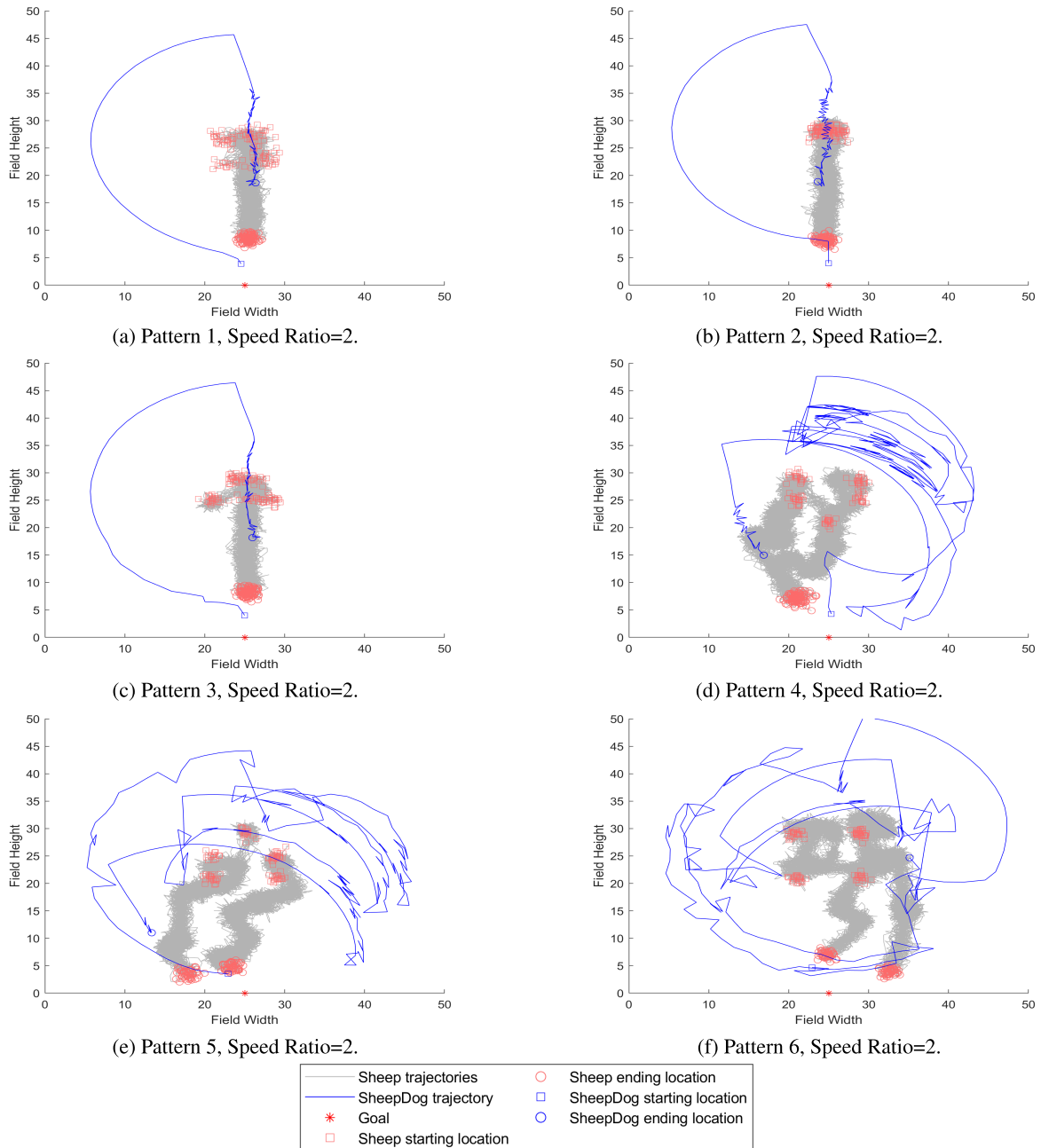


FIGURE 7. Footprint of the sheepdog and sheep in different environments and speed differentials.

two algorithms using the same configuration discussed in the experimental analysis. We limited the comparison to a single speed differential ratio of 1.25, and a single spatial distribution “P1”. We varied the number of sheep agents from 10 to 100 in a step of 10, and investigated two different values for the sheep’s neighborhood in Strömbom’s model: $\frac{1}{2}N$, and $\frac{3}{4}N$.

Figures 5a and 5b were obtained using 30 runs per each combination. It is demonstrated that the new model outperforms Strömbom’s model. The proposed model was able to complete all the tasks in less than 200 steps. On the other hand, Strömbom’s model failed to complete

the tasks in most of the runs up to 2000 steps as shown in figure 5b.

B. SINGLE SHEEPDOG COMPLEXITY

Due to the limitations of Strömbom’s model shown earlier, we continued the analysis with the proposed model. The average completion time over 30 runs ranged from 100 to 350 at sheepdog-to-sheep speed ratio of 1, and from 59 to 129 at a ratio of 3. We fit a linear and a parabolic function, as follows:

$$y_1 = a_1 \times x + a_2$$

$$y_2 = a_3 \times (x - a_4)^2 + a_5$$

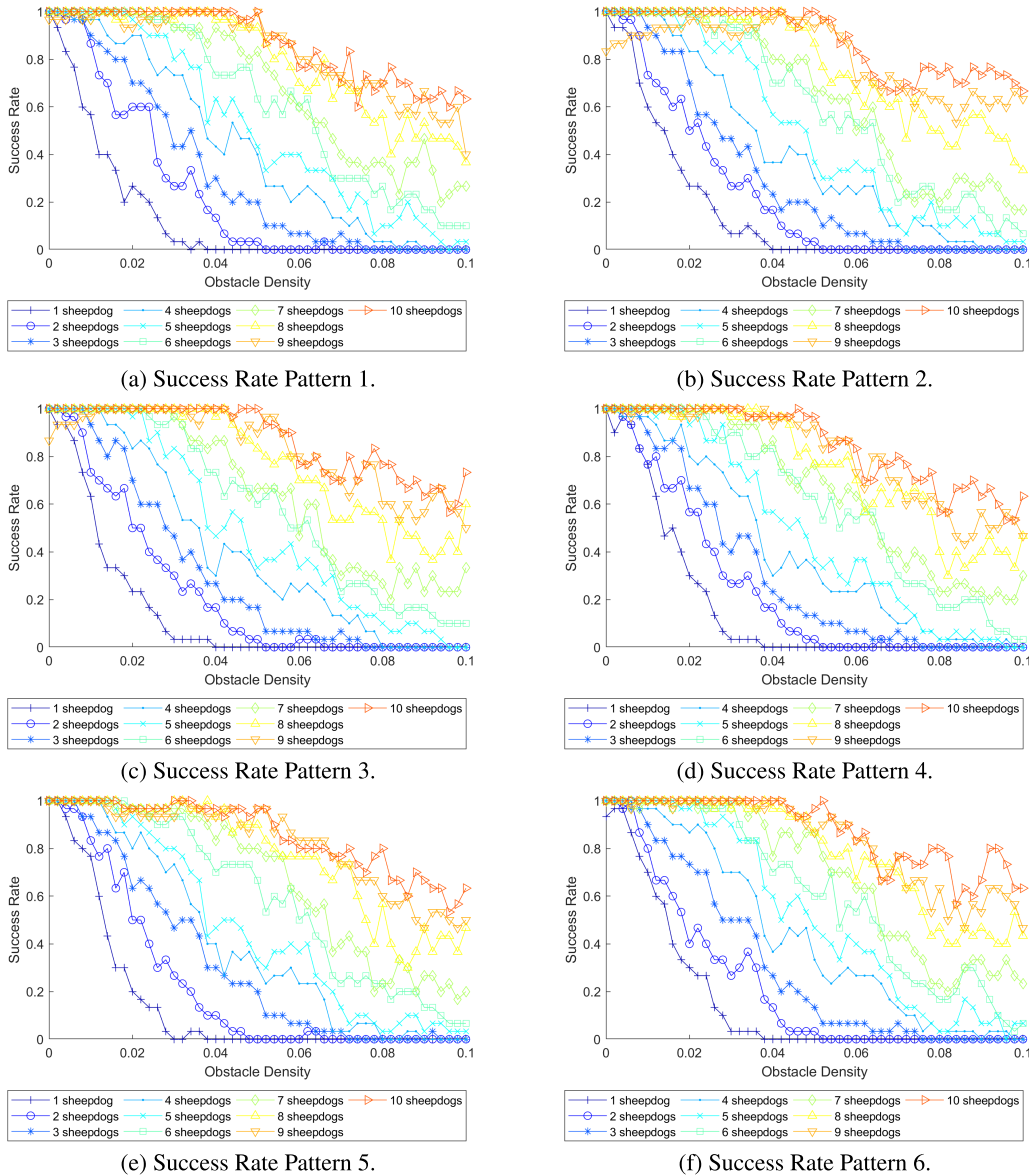


FIGURE 8. Success rate Vs obstacle density.

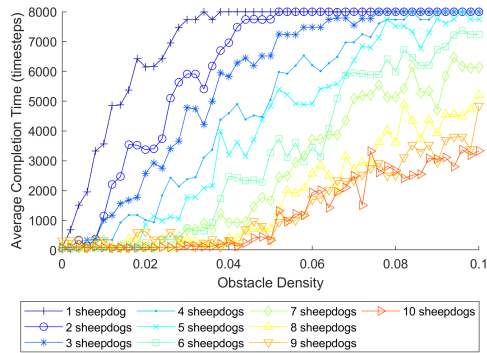
TABLE 3. Root Mean Square Error (RMSE) and parameters for the linear and parabolic curves in each environment.

Env	Linear			Parabolic			
	RMSE	a_1	a_2	RMSE	a_3	a_4	a_5
P1	25.99	-18.18	108.01	10.06	11.96	2.76	60.56
P2	30.86	-22.71	124.7	7.16	14.99	2.76	65.42
P3	27.64	-20.49	117.38	8.52	13.13	2.78	63.82
P4	106.13	-72.74	290.76	70.22	39.74	2.92	98.08
P5	87.49	-64.53	280.16	75.35	22.20	3.45	96.44
P6	180.29	-70.97	326.57	146.52	52.45	2.68	142.28

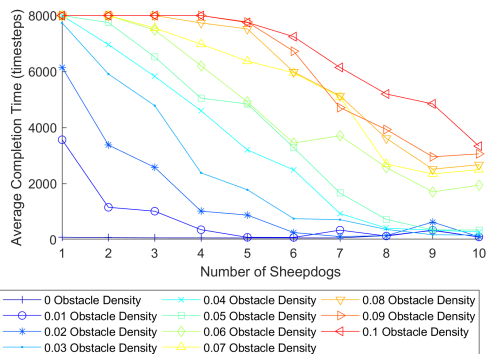
where y_1 is the linearly predicted average completion time, y_2 is the quadratic predicted average completion time, and x is the sheepdog-sheep-ratio. The parameters of the two equations for each of the six environments are shown in Table 3. The quadratic fit consistently yielded a better root mean square error (RMSE) compared to the linear fit.

The effect of the spatial distribution of the sheep at the time of initialization is apparent from the fitted functions. The first three environments $P1$ to $P3$ have similar coefficients. The last three have significantly different larger coefficients, and larger intercepts.

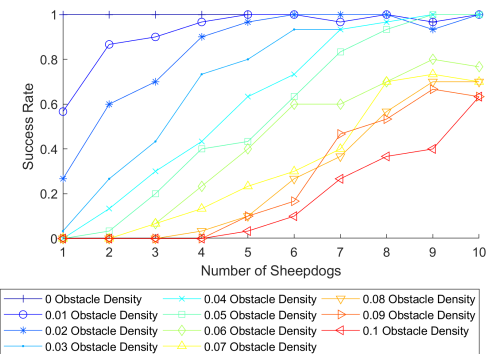
The effect of the spatial distributions on completion time is bipolar, where the first three scenarios display lower average and variance of the completion time over the 30 runs. Figure 6a shows a boxplot of the dispersion index over all sheep. When we zoom on the top 25% and 5% in Figures 6b and 6c, respectively, the differences between $P2$, on the one hand, and $P1$ and $P3$, on the other hand, becomes apparent. The high density initialization of the sheep in $P2$ shows smaller dispersion over the whole run; thus, it indicates that the movements of the sheepdog did not cause the sheep to disperse, and instead, maintained their grouping



(a) Completion Time vs Obstacle Density.



(b) Completion Time vs Number of Sheepdogs.



(c) Success Rate Vs Number of Sheepdogs.

FIGURE 9. Completion time and success rate under Pattern 1 spatial distribution.

together due to the simple circular path design we have adopted.

To understand the phenotypic characteristics of the trajectories taken by the sheep and shepdog, a sample of the simulations spanning different shepdog-to-sheep speed ratio and spatial distributions at the initialization phase are displayed in Figures 7a to 7f. The trajectories of the shepdog become much wider when the initialization of the sheep is sparse. We also found differences in these footpaths when the speed varies in the same environments. These differences are negligible in the first three environments, and become more significant in the last three.

C. MULTIPLE SHEPDOG COMPLEXITY

Figure 8 shows the success rate versus obstacle density for 1 to 10 shepdogs for the 6 different sheep initialization

TABLE 4. Thresholds for obstacle densities to achieve targeted success rates per number of sky shepherds.

M	Success rate			
	100%	$\geq 90\%$	$\geq 50\%$	$> 0\%$
1	0.0%	0.2%	1.0%	3.2%
2	0.2%	0.8%	2.4%	5.0%
3	0.4%	1.0%	2.8%	7.4%
4	0.4%	1.4%	3.6%	8.2%
5	1.8%	2.8%	4.6%	9.4%
6	2.2%	3.6%	6.2%	$> 10\%$
7	0.6%	4.4%	6.6%	$> 10\%$
8	1.4%	5.0%	8.0%	$> 10\%$
9	-	5.6%	9.8%	$> 10\%$
10	4.4%	5.0%	$> 10\%$	$> 10\%$

patterns. Varying the initialization of sheep location/distribution makes little difference to the success rate, but some general trends are evident.

Firstly, an increase in obstacle density increases the complexity of the task as demonstrated by the decline in success rate. Secondly, an increase in the number of dogs improves the success rate in most cases. The interaction of these two influencing factors is also visible in the graphs. For instance, in Figure 8a, it can be seen that the maximum density above which all 30 runs were unsuccessful was 3.4% for 1 shepdog, 6.8% for 2 shepdogs, 7.6% for 3 shepdogs and 9% for 4 shepdogs. Five Shepdogs failed in all 30 trials at a density of 9.6% and 6 or more shepdogs had success in 1 or more trials for all object densities between 0 and 0.1.

Continuing our analysis of the individual graphs in Figure 8, one can observe that if linear lines of best fit are used to represent the data, the gradients tend to decrease for all initialization patterns, as the number of shepdogs increases. This indicates that an increase in resources is appropriate to tackle the rise in problem complexity offered by increasing obstacle density. Analysing completion time, Figure 9a shows that if an average completion time of 2000 steps is required, this can be met by 10 shepdogs in environment densities of up to $\approx 7\%$, whilst five and one dog(s) achieve this completion time for obstacle densities of less than $\approx 3\%$ and $\approx 1.0\%$ respectively. This gain in performance provides an interesting trade-off between the cost of resources (number of shepdogs) and the corresponding benefits gleaned (higher success rates).

Inspecting the results for initial sheep distribution, Pattern P1 in Figure 9a, it is noticed that there is a phase transition in the gradient of the plots for 1-3 and 8-10 shepherds at an obstacle density of ≈ 0.04 . If one considers splitting the graph (and its associated data) at this point and provide a piece-wise linear fit comprising two lines, one for the obstacle density in the interval $[0,0.04]$, and the second for obstacle densities in the interval $(0.04,1]$, we can see that for the region $[0,0.04]$ where the problem is solvable independent of the number of shepdogs, the gradients decrease as the number of shepdogs increase. Within the obstacle density range range $(0.04,1]$,

this is again true where the problem is solvable, namely for 4 or more sheepdogs. Given the linear nature of these graphs, this suggests that the use of additional sheepdogs overcomes the increase in complexity due to an increase in obstacle density. However, we note that there is a diminishing return on the addition of sheepdogs; we contend this is, in part, brought about by their interaction with the obstacles. The phase transition does not occur due to change in number of sheepdogs alone. Due to lack of coordination and planning mechanisms among the agents who are moving in a reactive manner, the increase in obstacle densities cause the sheepdogs to disconnect and potentially work in conflict to each other.

VI. CONCLUSION AND FUTURE WORK

In this work, we identified influencing factors that impact the complexity of the swarm guidance problem using a shepherding approach. In the case of a single shepherd, we examined the relationship between the relative speed of the sheepdog to the sheep and the spatial density of the sheep during initialization. Both influencing factors were found to compound the task complexity as indicated by a decline in success rate. We further extended the analysis to study the interaction between the number of sheepdogs and the density of obstacles in the environment. The analysis revealed a phase transition as we move from a lower number of sheepdogs (< 4) to a larger number of sheepdogs (≥ 4) and as the density of obstacles increased. The phase transition is specifically caused by the interaction between the number of sheepdogs and density of obstacles in the environment; it is not solely due to a change in the number of sheepdogs used. Due to lack of coordination and planning mechanisms among the agent who are moving in a reactive manner, the increase in obstacle densities cause the sheepdogs to disconnect and potentially work in conflict to each other. These findings suggest that the complexity for the swarm guidance task increases as the (1) the relative speed of the sheepdog to the sheep decreases; (2) the dispersion of sheep at the start of the task increases; (3) the number of sheepdogs decreases; and (4) the density of obstacles in the environment increases.

We found a linear relationship between the completion time and an increase in obstacle density, and noted completion time growth reducing with the introduction of additional sheepdogs. We, therefore, conclude that the use of additional sheepdogs overcomes the increase in complexity due to the rise in obstacle density.

With the work in this study presuming a specific sheepdog formation, future work will examine the benefit of optimising sheepdog placement dynamically to create adaptive formations in the hope of better tackling the complexity inherent in the shepherding problem. We will also explore federated learning as a mechanism to allow privacy in the exchange of positional information for the sheep [34], [35].

ACKNOWLEDGMENT

Dr. D. Strömbom for providing them with a copy of his code.

REFERENCES

- [1] M. Bedau, "Artificial life: Organization, adaptation and complexity from the bottom up," *Trends Cognit. Sci.*, vol. 7, no. 11, pp. 505–512, Nov. 2003.
- [2] R. E. Wood, "Task complexity: Definition of the construct," *Organizational Behav. Human Decis. Processes*, vol. 37, no. 1, pp. 60–82, Feb. 1986.
- [3] T. Alpcan, T. Everitt, and M. Hutter, "Can we measure the difficulty of an optimization problem?" in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 356–360.
- [4] D. E. Goldberg, J. rey Horn, and K. Deb, "What makes a problem hard for a classier system," Illinois Genetic Algorithms Lab., Dept. Gen. Eng., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, IlliGAL Rep. 92007, 1992.
- [5] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, Mar. 2013.
- [6] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, Aug. 2018.
- [7] N. K. Long, K. Sammut, D. Sgarioto, M. Garratt, and H. A. Abbass, "A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 4, pp. 523–537, Aug. 2020.
- [8] H. Abbass and R. Hunjet, "Smart shepherding: Towards transparent artificial intelligence enabled human-swarm teams," in *Shepherding UxVs for Human-Swarm Teaming: An Artificial Intelligence Approach*. Cham, Switzerland: Springer, 2020, pp. 1–27.
- [9] D. Strömbom, 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," *J. Roy. Soc. Interface*, vol. 11, no. 100, Nov. 2014, Art. no. 20140719.
- [10] M. Isobe, D. Helbing, and T. Nagatani, "Experiment, theory, and simulation of the evacuation of a room without visibility," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066132.
- [11] P. Nalepka, R. W. Kallen, A. Chemero, E. Saltzman, and M. J. Richardson, "Herd those sheep: Emergent multiagent coordination and behavioral-mode switching," *Psychol. Sci.*, vol. 28, no. 5, pp. 630–650, May 2017.
- [12] 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 Trans. Robot.*, vol. 34, no. 4, pp. 901–915, Aug. 2018.
- [13] N. M. P. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *J. Hazardous Mater.*, vol. 154, nos. 1–3, pp. 880–887, Jun. 2008.
- [14] D. Cohen, "Cellular herding: Learning from swarming dynamics to experimentally control collective cell migration," in *Proc. APS Meeting Abstr.*, 2019.
- [15] T. Miki and T. Nakamura, "An effective simple shepherding algorithm suitable for implementation to a multi-mmobile robot system," in *Proc. 1st Int. Conf. Innov. Comput., Inf. Control (ICICIC)*, vol. 3, Aug/Sep. 2006, pp. 161–165.
- [16] K. Yaxley, N. McIntyre, J. Park, and J. Healey, "Sky shepherds: A tale of a uav and sheep," in *Shepherding UxVs for Human-Swarm Teaming: An Artificial Intelligence Approach*. Cham, Switzerland: Springer, 2020.
- [17] H. T. Nguyen, T. D. Nguyen, M. Garratt, K. Kasmari, S. Anavatti, M. Barlow, and H. A. Abbass, "A deep hierarchical reinforcement learner for aerial shepherding of ground swarms," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2019, pp. 658–669.
- [18] R. S. MacKay, "Nonlinearity in complexity science," *Nonlinearity*, vol. 21, no. 12, pp. T273–T281, Dec. 2008.
- [19] Y. Sueoka, M. Ishitani, and K. Osuka, "Analysis of sheepdog-type robot navigation for Goal-Lost-Situation," *Robotics*, vol. 7, no. 2, p. 21, May 2018.
- [20] S. Razali, Q. Meng, and S.-H. Yang, "Immune-inspired cooperative mechanism with refined low-level behaviors for multi-robot shepherding," *Int. J. Comput. Intell. Appl.*, vol. 11, no. 01, Mar. 2012, Art. no. 1250007.
- [21] J. Brulé, K. Engel, N. Fung, and I. Julien, "Evolving shepherding behavior with genetic programming algorithms," 2016, *arXiv:1603.06141*. [Online]. Available: <http://arxiv.org/abs/1603.06141>
- [22] W. Lee and D. Kim, "Autonomous shepherding behaviors of multiple target steering robots," *Sensors*, vol. 17, no. 12, p. 2729, Nov. 2017.
- [23] E. Maschian and M. Royan, "Cooperative control of a multi robot flocking system for simultaneous object collection and shepherding," in *Computational Intelligence*. Cham, Switzerland: Springer, 2015, pp. 97–114.

- [24] K. Fujioka, "Effective herding in shepherding problem in V-formation control," *Trans. Inst. Syst., Control Inf. Engineers*, vol. 31, no. 1, pp. 21–27, 2018.
- [25] M. H. Linder and B. Nye, "Fitness, environment and input: Evolved robotic shepherding," Dept. Comput. Sci., Swarthmore College, Swarthmore, PA, USA, Tech. Rep., 2010.
- [26] J.-M. Lien, S. Rodriguez, J. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 3402–3407.
- [27] S. Kalantar and U. Zimmer, "A formation control approach to adaptation of contour-shaped robotic formations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 1490–1497.
- [28] B. Bat-Erdene and O.-E. Mandakh, "Shepherding algorithm of multi-mobile robot system," in *Proc. 1st IEEE Int. Conf. Robotic Comput. (IRC)*, Apr. 2017, pp. 358–361.
- [29] T. Miki and T. Nakamura, "An effective rule based shepherding algorithm by using reactive forces between individuals," *Int. J. Innov. Comput., Inf. Control*, vol. 3, no. 4, pp. 813–823, 2007.
- [30] T. Nguyen, J. Liu, H. Nguyen, K. Kasmarik, S. Anavatti, M. Garratt, and H. Abbass, "Perceptron-learning for scalable and transparent dynamic formation in Swarm-on-Swarm shepherding," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020.
- [31] A. Clapham, "Over-dispersion in grassland communities and the use of statistical methods in plant ecology," *J. Ecol.*, vol. 24, no. 1, pp. 232–251, 1936.
- [32] J. Pery and R. Mead, "On the power of the index of dispersion test to detect spatial pattern," *Biometrics*, vol. 35, no. 3, pp. 613–622, 1979.
- [33] R. Fernandez Rojas, E. Debie, J. Fidock, M. Barlow, K. Kasmarik, S. Anavatti, M. Garratt, and H. Abbass, "Electroencephalographic workload indicators during teleoperation of an unmanned aerial vehicle shepherding a swarm of unmanned ground vehicles in contested environments," *Frontiers Neurosci.*, vol. 14, Feb. 2020, Art. no. 40.
- [34] M. Chen, Z. Yang, W. Saad, C. Yin, H. Vincent Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," 2019, *arXiv:1909.07972*. [Online]. Available: <http://arxiv.org/abs/1909.07972>
- [35] Z. Yang, M. Chen, W. Saad, C. Seon Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," 2019, *arXiv:1911.02417*. [Online]. Available: <http://arxiv.org/abs/1911.02417>



SABER ELSAYED (Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales (UNSW) Canberra, Australia, in 2012. He is currently a Senior Lecturer with the School of Engineering and Information Technology, UNSW Canberra. His research interests include evolutionary computation and its application. He is also the Chair of the IEEE Computational Intelligence Society (ACT Chapter).



ANTHONY PERRY received the bachelor of surveying degree from the University of South Australia, in 1994, the graduate diploma degree in ecology from The University of Adelaide, in 1995, and the M.Sc. degree in operations research and statistics from the University of New South Wales (UNSW) Canberra, in 2000. He is currently working with the Advanced Vehicle Systems team, where he is also working on aspects of navigation, distributed contextual awareness, and swarm dynamics. His current research interests include the connection between shepherding and control problems for remote and distributed agents.



HEMANT KUMAR SINGH (Senior Member, IEEE) is currently a Senior Lecturer with the School of Engineering and Information Technology, University of New South Wales (UNSW) Canberra, Australia. His research interests include evolutionary computation methods for optimization and decision-making. Within this domain, he has investigated various aspects, such as constraint handling, multiple conflicting criteria, robustness, and hierarchical objectives. He is also the Co-Chair of IEEE task-force on bilevel optimization.



HEBA EL-FIQI (Member, IEEE) received the M.Sc. degree in computer science from Cairo University, in 2008, and the Ph.D. degree in computer science from the University of New South Wales (UNSW) Canberra, Australia, in 2013. She is currently a Senior Research Fellow with the School of Engineering and Information Technology, UNSW Canberra. Her research interests include swarm intelligence, artificial intelligence, brain-computer interface, machine learning, computational linguistics, computer networks, and security systems.



BENJAMIN CAMPBELL received the bachelor's degree (Hons.) in electrical and mechatronic engineering from the University of South Australia, in 2006. He was employed with BAE Systems Australia, from 2007 to 2009. Since 2010, he has been employed as a Vehicle Electronic Systems Researcher with the Defence Science and Technology Group. His current research interests include the distributed control of vehicle mission systems with a focus on context awareness and distributed control of RF communication signatures.



ROBERT HUNJET is currently the Group Leader of the Advanced Vehicle Systems Science and Technology Capability, Defence Science and Technology Group, Australia, and an (Adjunct) Associate Professor with the University of New South Wales (UNSW) Canberra. His research interests include wireless network performance, tactical network survivability, distributed control and the use of self-organisation, swarming, and emergence to enable autonomy in multi-unmanned system to (multi) human teams.



HUSSEIN A. ABBASS (Fellow, IEEE) is currently a Professor with the University of New South Wales (UNSW) Canberra, Australia. His current research interests include trusted human-swarm teaming with an aim to design next generation trusted and distributed artificial intelligence systems that seamlessly integrate humans and machines. He was the Vice-President for Technical Activities for the IEEE Computational Intelligence Society, from 2016 to 2019. He is also the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE.

...