Title	Swarm Shepherding with Multiple Steering Agents in Limited Information Environments
Author(s)	Li, Aiyi
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/101773
rights	
Note	

Swarm Shepherding with Multiple Steering Agents in Limited Information Environments

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2025

Aiyi LI

List of Publications

Journal Papers

1. **A. Li**, M. Ogura and N. Wakamiya, "Swarm shepherding using bearing-only measurements," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, accepted for publication.

2. **A. Li**, M. Ogura and N. Wakamiya, "Communication-free shepherding navigation with multiple steering agents," *Frontiers in Control Engineering*, vol. 4, p. 989232, 2023.

International Conference Papers

3. **A. Li**, M. Ogura and N. Wakamiya, "Proposal of a bearing-only shepherding algorithm with limited sensing capabilities," in *the 28th International Symposium on Artificial Life and Robotics, the 8th International Symposium on BioComplexity, and the 4th International Symposium on Swarm Behavior and Bio-Inspired Robotics (AROB-ISBC-SWARM 2023)*, pp. GS34-1, 2023.

4. **A. Li**, M. Ogura and N. Wakamiya, "Proposal of farthest-agent targeting algorithm with indirect chasing," in *SICE Annual Conference 2022*, pp. 92-94, 2022.

Oral Presentations

5. **A. Li**, R. Oshima, Y. Murakami, M. Ogura, and N. Wakamiya, "Divide-and-rule strategies for shepherding swarms with inherent noise," in *the 11th SICE*

Multi-Symposium on Control Systems, pp. 1A7-2, 2024.

6. **A. Li**, M. Ogura and N. Wakamiya, "A distributed approach for shepherding with multiple steering agents," in *the 34th SICE Symposium on Decentralized Autonomous Systems*, pp. 2B2-3, 2022.

Preface

This dissertation draws inspiration from the remarkable self-organizing behaviors seen in nature, such as flocking in birds, schooling in fish, and guiding in animals. These natural phenomena demonstrate how complex, coordinated group movements emerge from simple, local interactions among individuals without centralized control. By translating these principles into engineered systems, this dissertation investigates swarm guidance approaches that leverage decentralized control and information-limited strategies to enable steering agents to guide groups of passive agents. These methods have broad applications, from environmental monitoring and precision agriculture to search and rescue.

Inspired by natural behaviors such as sheepdogs guiding sheep, ”shepherding” in swarm robotics involves a class of steering agents that influence another group of agents. Shepherding systems rely on steering agents to guide or repel passive members, thus directing the entire swarm toward a target.

Conventional approaches to shepherding employ various methods, including control theory, rule-based strategies, and reinforcement learning. Nevertheless, these methods often rely on continuous access to detailed information, such as centralized control for the steering agents or comprehensive sensing data about other agents. In practical scenarios, such assumptions are frequently unrealistic due to sensing limitations and communication constraints, which significantly reduce the feasibility of the aforementioned methods.

This dissertation presents shepherding algorithms that utilize limited information to enable cooperation among steering agents. The research aim is to find essential information types for accomplishing the shepherding task by proposing and evaluating practical algorithms. These algorithms emphasize simplicity by reducing reliance on complex data collection and extensive agent-to-agent communication, thereby enabling flexibility and scalability in managing larger swarms. Specifically, this research focuses on two key studies, each addressing distinct aspects of information constraints.

The first study employs a fully decentralized method in which each steering agent operates based solely on its local observations without any inter-agent communication. This design enables steering agents to achieve emergent coordination, as collective behavior arises naturally from the interplay of individual decisions. Specifically, each steering agent guides the swarm by independently selecting, driving, and switching target agents while maintaining a safe distance from other steering agents. Numerical simulations demonstrate that this communication-free method is scalable and robust under various conditions. In this context, "scalability" refers to the ability to manage larger swarm sizes and varied initial placements of agents without significant increases in computational or communication demands, and "robustness" denotes the ability to maintain effective guidance even when we change agent model parameters or fluctuations in sensing accuracy.

The second study introduces a bearing-only method that operates under stricter sensing limitations. In this context, each agent senses only the directions of neighboring agents without access to proximity or distance measurements. The algorithm utilizes directional information relative to the swarm and the goal to guide the movement of a steering agent. Additionally, to achieve coordination and prevent collisions under these constraints, the algorithm incorporates limited inter-agent communication, allowing each steering agent to share essential directional information about the guiding swarm. This low-level communication approach enables agents to adapt to scenarios involving multiple swarms. Simulations demonstrate the performance of this algorithm in guiding multiple swarms with multiple agents in various initial placements. The shepherding task generally remains successful even as the accuracy of bearing measurements for each steering agent decreases.

Through extensive simulations, this dissertation validates the proposed methods under two distinct settings to examine the limits in swarm control: (1) relative sensing without communication and (2) further reduced sensing with limited communication. Although intermediate configurations may be possible, focusing on these extreme scenarios provides critical insights into the information required for coordination. The results offer practical and quantitative benchmarks, specifying the sensing and communication limits that allow autonomous systems to operate reliably with limited data in real-world conditions. By defining these criteria, this research addresses pressing challenges in swarm robotics and multi-agent systems and establishes a foundation for future developments in adaptive, information-efficient swarm technologies.

In summary, this study advances our understanding of decentralized shepherding-type swarm control by utilizing the limited information requirements for coordination. These findings bridge theory with application, demonstrating that simple, rule-based interactions can produce scalable, effective swarm behaviors under realistic constraints. The solutions presented underscore the importance of simplicity and adaptability in engineered systems, paving the way for robust and resource-efficient swarm technologies that can perform in challenging, information-limited environments. The results of this study bring us closer to realizing the potential of swarms as intelligent, adaptive systems, poised to transform domains such as autonomous navigation, environmental management, and collaborative robotics.

Acknowledgement

I would like to express my sincere gratitude to all the members of the Bio-inspired Networking Laboratory at Osaka University. I am especially grateful to my advisor, Guest Professor Masaki Ogura, for his invaluable guidance and support. Prof. Ogura has been my advisor since I joined the laboratory as a research student in 2020, continuing through the master's program and now the PhD program. Over these years, I have gradually developed my skills in academic research and problem-solving. I am deeply thankful to Professor Naoki Wakamiya for his dedicated management of the laboratory and his valuable suggestions on my research theme and progress. I am also very grateful to Prof. Takeshi Hirai and Prof. Yang Bai as researchers for their advice in building my career path. Additionally, I am thankful to all other members of this laboratory, including current and former students, for their companionship and support throughout my research journey.

I am fortunate to have joined the Humanware Innovation Program at Osaka University. The collaborative research in this program, conducted with other students, has provided me with valuable practical experiences and opportunities to undertake challenging experiments involving swarm systems.

Personally speaking, I am especially grateful to my partner and friends for engaging with me on my research theme and progress. Communicating my research to others has been essential in deepening my understanding, shaping my vision, and helping me recognize the academic and industrial value of my work.

Once again, I am thankful to everyone who has supported me throughout these years of research. I am truly thankful for the opportunity to work on topics related to swarm systems and swarm control, which have been fascinating areas of study.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Self-organized group formation and collective motion in the form of swarms are hallmarks of living systems over a wide range of length scales, from microorganisms to fish schools, bird flocks, and animal herds [1, 2, 3]. These behaviors are remarkable in that they emerge without any centralized coordination, relying instead on local interactions among individual members of the group [4, 5, 6]. Through these interactions, organisms collectively adjust their movements in response to environmental cues and the positions of neighbors, producing cohesive formations and cooperative behaviors.

The study of swarm behaviors has attracted interest in the field of biology and inspired the development of artificial systems that utilize numerous agents to perform intelligent tasks. In swarm robotics, autonomous agents collaborate to complete complex objectives [7]. This capability supports a range of dynamic tasks, including shape assembly [8], coordinated drone flying [9], and control of magnetic microrobot systems [10, 11]. Such engineered swarms have found applications across diverse areas, including environmental monitoring, search and rescue, and precision agriculture, where multiple agents are deployed in various formations to perform intelligent tasks with flexibility.

However, in many circumstances, directly manipulating swarm behaviors or movements has significant difficulties and limitations due to the inherent autonomy and decentralized decision-making processes within these systems, highlighting the need for developing swarm control through external influences [12]. In biological swarms, individuals act independently and often respond unpredictably to stimuli,

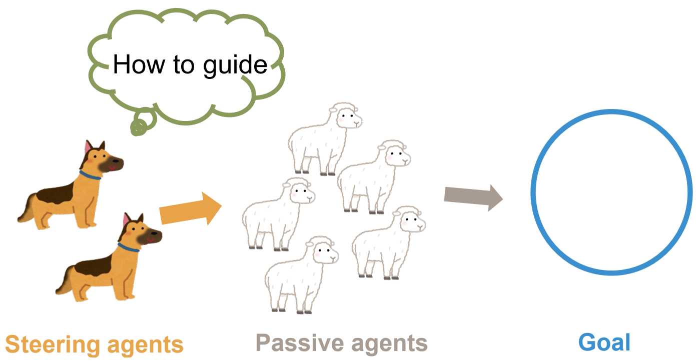


Figure 1. Illustration of the shepherding task for steering agents navigating a swarm of passive agents.

making direct command nearly impossible. Similarly, in artificial swarms, while robots can be programmed to localize and perform specific tasks, internal coordination becomes increasingly difficult as swarm size grows. Therefore, establishing scalable and smooth coordination within swarms is essential through swarm control.

A well-known example in swarm control and multi-agent systems is the leader-follower formation [13, 14]. This approach is widely applied in scenarios where structured movement is essential. In leader-follower formation, a designated leader directs followers toward a common goal, with each follower adjusting their actions based on the leader's movements. Although this method has shown effectiveness in certain controlled settings, the success of this method relies on each follower actively coordinating to avoid collisions and maintain formation [15]. However, achieving this level of coordination becomes increasingly difficult in large swarms or when agents operate with only localized information, making it challenging to anticipate the actions of distant agents.

On the other hand, a paradigmatic example of swarm control is the shepherding problem [16, 17], which is inspired by the natural behavior of sheepdogs herding sheep. This problem involves designing algorithms for shepherd-like steering agents to guide sheep-like passive agents (commonly referred to as swarms) toward a designated goal using repulsion forces, as illustrated in Figure 1. Beyond its biological inspiration, the shepherding study encompasses theoretical and practical significance. Theoretically, this research provides a unique framework to explore distributed control strategies, coordination mechanisms, and emergent behaviors in multi-agent systems [18]. It also contributes to understanding how local interactions lead to global outcomes, a central theme in swarm intelligence and control theory [19]. Practically, the applications of shepherding span diverse domains. In agriculture, robotic systems can be deployed for livestock herding [20, 21], reducing the dependency on human labor. In urban settings, shepherding-inspired algorithms have been used for crowd control and evacuation management [22], addressing challenges in public safety. Moreover, the coordination of large-scale robotic and microrobotic swarms [23], as in environmental monitoring, disaster response, and targeted drug delivery [24, 25], further underscores the practical relevance of this research. The overarching goal is to bridge the gap between understanding natural collective behaviors and engineering artificial swarm systems capable of tackling complex, real-world challenges.

The study of shepherding focuses on designing movement algorithms for steering agents, commonly referred to as shepherding algorithms. These algorithms can generally be classified into three categories; however, limited-information scenarios are rarely given priority in any of these categories.

The first category consists of control-theoretic methods, where researchers apply control theory to design precise movements for steering agents, guiding the swarm based on specific models and parameters [26, 27, 28, 29, 30, 31]. Although control-theoretic methods achieve a high degree of precision, reliance on predefined kinematic models and parameters limits their practicality when these underlying assumptions no longer hold [32].

The second category includes rule-based methods, which rely on relatively simple behavioral rules to direct and gather sheep-like agents toward a target [33, 34, 35, 36, 37]. Rule-based models offer a simpler and more adaptable approach, especially in situations where precise control of individual agents is neither feasible nor required. This adaptability makes rule-based algorithms particularly promising for real-world applications. Nevertheless, most rule-based approaches still assume considerable sensing and communication capabilities, which may be impractical in certain scenarios [38].

The third category encompasses learning or optimization-based path planning methods, which employ reinforcement learning or optimization techniques to help steering agents navigate effectively through environments with obstacles [39, 40]. These methods are particularly valuable in diverse environments with varying obstacles and conditions, where agents need to adapt flexibly. However, to simplify the learning process, these approaches often assume the sheep-like agents act as a single cohesive group, removing the need to account for individual differences in behavior [41].

1.2 Objective

This dissertation aims to find the key information types required by steering agents to accomplish the shepherding task by proposing and evaluating practical algorithms. In this context, we categorize information into two primary types: sensing information, which relates to an agent's ability to detect and determine the relative positions of nearby agents, and communication information, which involves data exchange between agents to enable cooperative behavior. We use rule-based methods for this categorization, as rule-based algorithms provide consistent criteria for evaluating feasibility. Addressing the objective will lay a framework that clarifies how combinations of sensing and communication capabilities influence the success of swarm shepherding.

This investigation contributes to practical applications by examining how simulations based on real-world constraints reveal the requirements for information in swarm coordination. These findings provide insights into optimizing system design for applications where information is costly or difficult to obtain, such as autonomous drone navigation in areas with low connectivity, or search-and-rescue operations in dynamic and unpredictable terrains.

To achieve the aforementioned aims, we study swarm shepherding from the perspective of sensing and communication information required in rule-based methods.

In the first study [42], we examine the feasibility of guiding steering agents under the constraint that sensing is limited to relative distances and directions without relying on communication for coordination. The results indicate that direct communication between agents is not required, as effective coordination can still be achieved by agents adjusting their behavior based solely on relative distances to other agents. Furthermore, the proposed approach leverages emergent inter-agent repulsion and distributed target selection, enabling agents to collectively guide the swarm toward a target without explicit communication.

In the second study [43], we further reduce the available sensing information by limiting it to bearing-only measurements, where each agent can detect only the directions of neighboring agents without access to relative distances. These constraints make cooperative shepherding significantly more challenging, as agents lack precise location information to evaluate proximity. To compensate for this reduced sensing capability, we introduce a low level of communication that enables agents to share brief directional information. This limited communication allows agents to confirm relationships among multiple swarms and maintain cooperative movements with the swarms. By integrating this low-level communication with bearing-only measurements, we demonstrate that agents can still assess their orientations, coordinate, and guide multiple swarms toward a goal under severely constrained sensing and communication conditions.

1.3 Target Scenario

We begin by introducing the commonly used movement model for passive agents and the goal settings in the shepherding problem, followed by a notable movement algorithm for steering agents. The task of shepherding involves using these algorithms to guide passive agents from their initial positions to a designated goal. Each algorithm is designed based on the specific information scenario in the study. Specifically, we assume each agent has no area or volume to simplify the model, allowing us to focus on the movement algorithms for steering agents.

1.3.1 Sheep Model

We consider a scenario in which M steering agents are tasked with guiding N sheep. Initially, the N passive agents are grouped into several distinct swarms, each with its spatial arrangement. The M steering agents are individually positioned at specified locations around these swarms to facilitate the guidance process. The passive agents and steering agents are assumed to move dynamically on a two-dimensional plane $\mathbb{R}^2$ in discrete time. To denote the sets of passive agents and steering agents, we use the notation $[N] = \{1, 2, \dots, N\}$ and $[M] = \{1, 2, \dots, M\}$, respectively. For any $i \in [N]$ and $k \in [M]$, we use $p_i(t), u_i(t)$ and $q_k(t), v_k(t)$ to denote the position and velocity, respectively, of the ith passive agents and kth steering agents, respectively, at time t. The movements of the passive agent and steering agent are thus defined as follows:

$$p_i(t+1) = p_i(t) + u_i(t),$$
$$q_k(t+1) = q_k(t) + v_k(t).$$

We assume that each passive agent and steering agent recognizes other agents within limited sensing ranges by the positive values r and r', respectively. Accordingly, the sets of other passive agents and steering agents recognized by the passive agent i at time t are given by

$$\begin{aligned} \mathcal{N}_i(t) &= \{j \in [N] \mid 0 < \|p_i(t) - p_j(t)\| < r\}, \\ \mathcal{M}_i(t) &= \{k \in [M] \mid 0 < \|p_i(t) - q_k(t)\| < r\}, \end{aligned} \tag{1}$$

respectively. Similarly, the sets recognized by the kth steering agent at time t are given by

$$\mathcal{N}'_k(t) = \{j \in [N] \mid 0 < \|q_k(t) - p_j(t)\| < r'\},$$

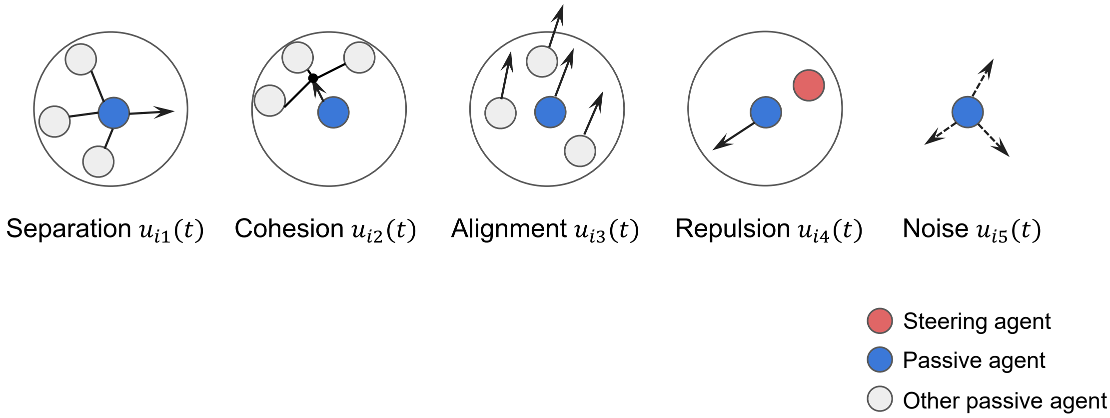


Figure 2. Schematic of the passive agent model described in Equation (3). A passive agent receives separation, alignment, cohesion from other sheep, and repulsion from steering agents within the sensing range. The passive agent is also exposed to noise.

$$\mathcal{M}'_k(t) = \{\ell \in [M] \mid 0 < \|q_k(t) - q_\ell(t)\| < r'\},$$

respectively.

Following the convention used for the boid model [5] and the shepherding problem [34, 35, 37], the movement of the ith passive agent at time t is defined by

$$u_i(t) = c_1 u_{i1}(t) + c_2 u_{i2}(t) + c_3 u_{i3}(t) + c_4 u_{i4}(t) + c_5 u_{i5}(t), \tag{2}$$

where $u_{i1}(t)$, $u_{i2}(t)$, and $u_{i3}(t)$ denote the forces of separation, cohesion, alignment, respectively, between sheep; $u_{i4}(t)$ denotes the force of repulsion from the steering agents; and $u_{i5}(t)$ denotes a uniformly distributed random vector representing noise. Meanwhile, c_1, c_2, c_3, c_4, and c_5 are positive constants. Specifically, we define the first four vectors as

$$
\begin{aligned}
u_{i1}(t) &= -|\mathcal{N}_i(t)|^{-1} \sum_{j\in\mathcal{N}_i(t)} \psi(p_j(t) - p_i(t)), \\
u_{i2}(t) &= |\mathcal{N}_i(t)|^{-1} \sum_{j\in\mathcal{N}_i(t)} \phi(p_j(t) - p_i(t)), \\
u_{i3}(t) &= |\mathcal{N}_i(t)|^{-1} \sum_{j\in\mathcal{N}_i(t)} \phi(u_j(t-1)), \\
u_{i4}(t) &= -|\mathcal{M}_i(t)|^{-1} \sum_{\ell\in\mathcal{M}_i(t)} \psi(q_l(t) - p_i(t))
\end{aligned}
\tag{3}
$$

where $\phi(x) = x/\|x\|$ denotes a normalization operator, to represent the direction of a vector; and

$$
\psi(x) = \begin{cases} x/\|x\|^3, & \text{if } \|x\| \geq \delta, \\ x/(\|x\|\delta^2), & \text{if } 0 < \|x\| < \delta, \\ 0, & \text{otherwise,} \end{cases} \tag{4}
$$

denotes a potential-like function, to represent the interference caused by the proximity between two agents. We set the constant δ to be greater than 1 to prevent the value of $\|\psi(x)\|$ from diverging when $\|x\|$ is less than 1. One example illustrating the value of $\psi(x)$ is visualized in Figure 3. Among the Equations in Equation (3), the function $\psi(x)$ allows both $u_{i1}(t)$ and $u_{i3}(t)$ to be calculated based on varying distance scales between each pair of sheep, helping to prevent collisions when passive agents are too close [44] and to avoid dispersion when distances exceed the sensing range r. Additionally, we set $u_2(t)$ to a time-delay term to allow each agent to attempt to observe and follow the movement of the surrounded sheep.

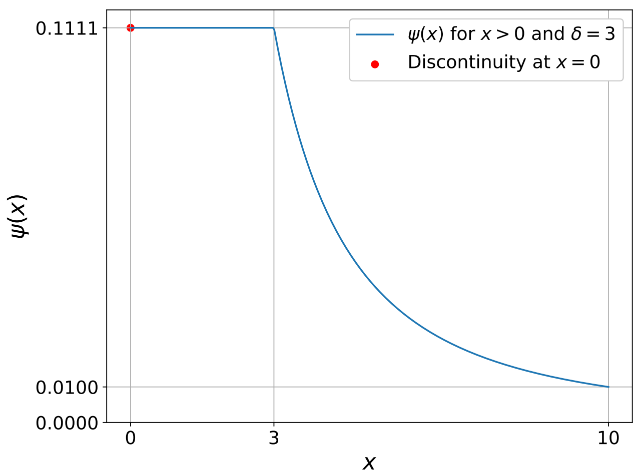


Figure 3. Function $\psi(x)$ defined in Equation (4) when $\delta = 3$. The range of x, representing the distance between two agents, is set to $x \geq 0$. The value of $\psi(x)$ is 0 at $\psi(0)$, reaches a threshold of $1/\delta^2$ when $0 < x \leq \delta$, and gradually decreases when $x \geq \delta$.

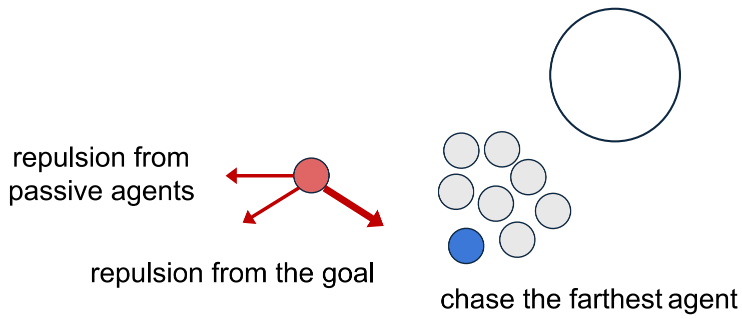


Figure 4. Illustration of the concept of the Farthest-Agent Targeting algorithm. A steering agent (red dot) chases the farthest agent from the goal as its target agent (blue dot).

1.3.2 Goal

The objective of the shepherding task is to guide all passive agents into a designated region called the goal. The task is considered complete only when all passive agents are within the goal area simultaneously at the same time step t. We let the goal region $G \subset \mathbb{R}^2$ be a closed disk with centre $x_g \in \mathbb{R}^2$ and radius $R_g > 0$. The value of goal radius R_g influences the results of shepherding because the larger the goal radius, the sooner the shepherding is completed. By contrast, the smaller the goal radius, the more time steering agents consume until all passive agents are guided into the goal region. Therefore, we set the goal radius R_g based on the length of the swarm shape when it reaches a relatively stable shape. In this state, the distances between passive agents remain almost constant after a few time steps since initialization in the preliminary simulation, without any interference from the steering agents. The specific values and adjustments for radius R_g and passive agent number N are introduced in the experiment section of each study.

1.3.3 Example of Shepherding Algorithms

Among the methods introduced in Section 1.1 for designing movements of steering agents, the Farthest-Agent Targeting (FAT) algorithm [37] stands out for its simplicity and effectiveness in scenarios involving a single steering agent (i.e., $M = 1$) guiding a swarm. In this algorithm, the steering agent primarily moves toward the position of the passive agent farthest from the goal, as observed by the steering agent as $\arg\max_{i \in \mathcal{N}(t)} \|p_i(t) - x_g\|$. An illustration is presented in Figure 4 and the detailed implementation of the algorithm is introduced in the following section. While the FAT algorithm provides a straightforward approach, its primary limitations include a lack of scalability due to reliance on a single steering agent and potential inaccuracies caused by the relative placements of steering agents and swarms. Despite these shortcomings, the FAT algorithm has inspired the development of numerous advanced algorithms that aim to overcome these issues.

1.4 Outline of Dissertation

The remainder of this dissertation is organized into two sections according to the objective introduced in Section 1.2. Chapter 2 presents the proposed communication-free shepherding algorithm, designed for scenarios with limited sensing and no communication capabilities. Chapter 3 describes the proposed bearing-only shepherding algorithm, developed for scenarios with further reduced sensing capabilities and limited communication options. Chapter 4 presents the conclusion of this dissertation.

Chapter 2

Shepherding Control by Communication-free Algorithm

2.1 Introduction

Our first study investigates whether communication between steering agents is necessary for successful shepherding. Previous studies often rely on communication between agents for the coordination of multiple steering agents, and various methodologies have been proposed to address challenges in information control. For example, navigation using steering agents in a prescribed formation demonstrates effective control [33]. Building on this concept, a 3-D guiding algorithm applies dimension reduction to manage the complexity of the multi-agent system [27]. Other approaches include caging-based algorithms designed for guiding a flock of agents [31, 29]. Additionally, centralized shepherding algorithms assign specific paths to each steering agent [38], and quasi-decentralized control laws using sliding mode control facilitate coordination among multiple steering agents [26].

Most existing shepherding algorithms with multiple steering agents assume the existence of a central coordinator [33, 27, 26, 29, 38, 31]. This assumption requires the coordinator to observe the whole system including the steering agents. However, these requirements can severely limit the practical feasibility of the algorithms. Although we can find in the literature a few decentralized shepherding algorithms with multiple steering agents, these works still implicitly assume the communication among steering agents. For example, the shepherding algorithm proposed by [36] requires that a steering agent can know the intention of another, which is hard to realize without communication between these agents. Also, in the shepherding algorithm developed by [30], the steering agents initially need to perform multiple rounds of communications for executing a distributed clustering algorithm to reach a consensus on which sub-swarm is shepherded by which steering agent.

The objective of this study is to propose an algorithm for communication-free shepherding navigation with multiple steering agents, relying solely on their ability to sense relative distances and directions to achieve cooperation. Our approach is to start from an existing single-steering agent algorithm called Farthest-Agent Targeting algorithm [37]. Leveraging on the simplicity of the algorithm, we then construct an algorithm for shepherding by multiple steering agents under the assumption that each steering agent knows its relative position to the goal and the relative position of other agents within the shepherd's recognition range. Within the proposed algorithm, although each steering agent attempts to guide the whole swarm by chasing its target passive agent independently and without inter-steering agent communication, cooperative behavior emerges as a consequence of the spatial distribution of steering agents induced by the inter steering agent repulsion built into the algorithm. The target passive agent of a steering agent is determined as the passive agent maximizing the weighted difference between the sheep's distance from the goal and the one from the shepherd. The improved performance of the proposed algorithm with an increasing number of steering agents is demonstrated through extensive numerical simulations.

The remainder of this chapter is organized as follows. Section 2.2 describes our proposed decentralized shepherding algorithm. Section 2.3 and Section 2.4 describe the baseline algorithms and centralized algorithms separately for comparison. Section 2.5 presents the numerical simulations and discusses the robustness of the proposed algorithm.

2.2 Proposed Algorithm

In this section, we describe the algorithm that we propose for the movement of the steering agents. We start by recalling the Farthest-Agent Targeting (FAT) algorithm [37] designed for the case of a single steering agent (i.e., $M = 1$). In the algorithm, the movement of the (1st) steering agent is specified as $q_1(t+1) = q_1(t) + v_1(t)$, where $v_1(t) \in \mathbb{R}^2$ represents the movement vector of the shepherd. Let us denote the position of the passive agent farthest from the goal by $\xi_1(t)$; i.e., define

$$\xi_1(t) = \operatorname*{arg\,max}_{p \in \{p_i(t)\}_{i \in [N]}} \|p - x_g\|.$$

Then, in the FAT algorithm, the movement vector $v_1(t)$ is specified as the weighted sum of the following three vectors:

$$\phi(\xi_1(t) - q_1(t)),\ -\psi(\xi_1(t) - q_1(t)),\ -\phi(x_g - q_1(t)), \tag{5}$$

which are, respectively, to realize the movement of the steering agent for chasing the farthest agent, taking an appropriate distance with the farthest agent, and pushing the farthest agent toward the goal region. As for the second term, the term allows us to realize an appropriate, non-vanishing distance for the same reason that the normalization ψ in the passive agent model allows a passive agent to avoid a collision. Despite being simple, the FAT algorithm is known for its effectiveness in performing the shepherding navigation with a single shepherd [37]. However, the algorithm requires knowledge of the positions of all sheep. Furthermore, when generalized to the situation of multiple steering agents, the formula would result in all the steering agents targeting the same sheep, which is presumably inefficient.

Based on these observations, in this study, we propose an extended version of the FAT algorithm to let each steering agent choose, as its target, a passive agent both close to itself and far from the goal. Specifically, we propose that the passive agent

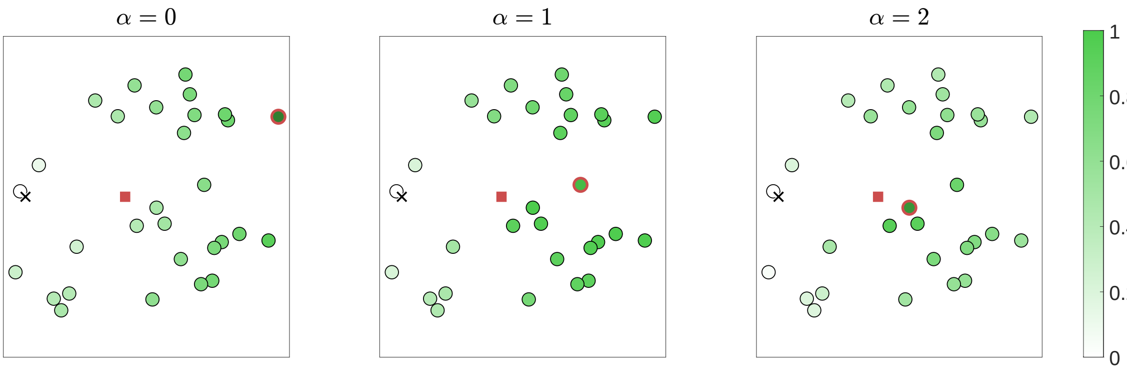


Figure 5. Target selection by the proposed algorithm. Goal: X; Sheep: circles; Shepherd: square. For each value of α, we color each sheep i using the $[0,1]$-normalized value of the objective function $\|p_i - x_g\| - \alpha\|p_i - q_k(t)\|$. When $\alpha = 0$, the steering agent targets the passive agent farthest from the goal. On the other hand, for larger α, the steering agent targets a passive agent far from the goal and close to the shepherd.

targeted by the kth steering agent is determined by the formula

$$\xi_k(t) = \operatorname*{arg\,max}_{p \in \{p_j(t)\}_{j \in \mathcal{N}'_k(t)}} \big(\|p - x_g\| - \alpha\|p - q_k(t)\|\big), \tag{6}$$

where $\alpha \geq 0$ is the parameter determining the behavior of steering agents within the proposed algorithm by balancing two factors. For example, when $\alpha = 0$, only the first term $\|p - x_g\|$ remains in the formula (6) and, therefore, all steering agents target the passive agent farthest from the goal; i.e., the proposed algorithm reduces to the FAT algorithm. On the other hand, when α is sufficiently large, each steering agent chooses the closest passive agent as its target, which specifically prevents the scattering phenomenon caused by the FAT algorithm, as illustrated in Figure 5. Hence, we can expect that choosing a moderate value of α would result in a control strategy that is as effective as the FAT algorithm and is less suffered from the scattering phenomenon. We here emphasize that $\xi_k(t)$ is decidable by the kth steering agent because the sheep's relative position to the goal is computable as $p_j(t) - x_g = (p_j(t) - q_k(t)) + (x_g - q_k(t))$.

We can now state the proposed movement algorithm of the steering agents. As in the FAT algorithm, we let

$$q_k(t+1) = q_k(t) + v_k(t),$$

where $v_k(t)$ denotes the movement vector of the kth shepherd. This vector is to be constructed as the weighted sum of the following four vectors. First, we define

$$v_{k1}(t) = \phi(\xi_k(t) - q_k(t))$$

for the kth steering agent to chase the target sheep. Secondly, in order to take an appropriate distance between the steering agent and sheep, we define

$$v_{k2}(t) = -|\mathcal{N}'_k(t)|^{-1} \sum_{j \in \mathcal{N}'_k(t)} \psi\big(p_j(t) - q_k(t)\big) \tag{7}$$

so that the kth steering agent receives repulsion force from all the neighboring sheep. Thirdly, to achieve guidance toward the goal region, we define the vector

$$v_{k3}(t) = -\phi(x_g - q_k(t))$$

by adopting (5). Finally, in order to avoid competition among steering agents for efficient guidance, we introduce the vector

$$v_{k4}(t) = -\|x_g - q_k(t)\| \, |\mathcal{M}'_k(t)|^{-1} \sum_{\ell \in \mathcal{M}'_k(t)} \psi(q_\ell(t) - q_k(t)), \tag{8}$$

which represents repulsion between steering agents. Because steering agents need to be relatively closer to each other at the final stage of the shepherding navigation, we introduce the weight term $\|x_g - q_k(t)\|$. Now, based on the four vectors introduced above, we define the movement vector of the kth steering agent as

$$v_k(t) = d_1 v_{k1}(t) + d_2 v_{k2}(t) + d_3 v_{k3}(t) + d_4 v_{k4}(t) \tag{9}$$

for positive constants d_1, d_2, d_3, and d_4. We do not include the additive noise term in the movement of steering agents.

In the experiments described in Section 2.5, we use the parameters $d_1 = 2$, $d_2 = 100$, $d_3 = 1$, $d_4 = 2$, and $r' = 300$. We set the parameter α in Equation (6) to $\alpha = 1$, as it provides a balanced and intuitive value for this trade-off.

Moreover, in this study, we do not aim to analytically establish the effectiveness of the shepherding algorithms. The major reason for this choice is its intrinsic difficulty arising from the nonlinearity of the Boid model. In fact, several existing works [see, e.g., 36, 27, 33, 31, 29, 38, 26] on the shepherding problem do not provide a mathematical proof for the performance of the proposed control methodologies. This tendency is a common practice in the field of swarm guidance, as the nonlinearity of the swarm model often makes it challenging to perform a meaningful mathematical analysis.

More importantly, let us discuss the communication requirements of the proposed algorithm and existing distributed shepherding algorithms with multiple steering agents [36, 30]. The proposed algorithm does not require communication between steering agents in the sense that each steering agent requires only its relative position with other steering agents, which can be achieved with its own sensing devices. On the other hand, as discussed in Section 1, the existing algorithms require communication between steering agents because each agent needs to understand the target or key factors influencing the movement of other steering agents. Specifically, the algorithm by [36] can require $O(n_f M^2)$ times of communications between steering agents at each time step, where n_f denotes the number of sub-flocks of sheep. Also, within the algorithm presented by [30], in order to execute a clustering algorithm for determining which sub-swarm is chased by which shepherd, $O(n_f M)$ times of communications needs to be periodically performed between steering agents.

2.3 Baseline Algorithms for Single Steering Agent

We describe notable baseline methods used for comparison in our numerical simulations. Similarly, the methods presented in this section are not designed for scenarios with multiple shepherds and do not rely on inter-shepherd communication. These baseline methods are utilized to evaluate the effectiveness of the proposed algorithm, particularly in scenarios involving multiple steering agents.

2.3.1 Farthest-Agent Targeting with Occlusion

The farthest-agent targeting algorithm with occlusion (FAT-OCC) [37] is also considered. This algorithm is identical to the FAT algorithm except that the vector $v_{k2}(t)$ in (7) is modified as

$$v_{k2}(t) = -|\mathcal{N}'_{k,\mathrm{occ}}(t)|^{-1} \sum_{j\in\mathcal{N}'_{k,\mathrm{occ}}(t)} \psi(p_j(t) - q_k(t)),$$

in which the set $\mathcal{N}'_{k,\mathrm{occ}}(t)$ represents the set of passive agents recognizable under occlusion and is constructed as follows. For each t, we first initialize $\mathcal{N}'_{k,\mathrm{occ}}(t) = \emptyset$. We then order the set $\mathcal{N}'_k(t)$ as $(i_1, \dots, i_{|\mathcal{N}'_k(t)|})$ in such a way that $\|p_{i_1}(t)\| \leq \|p_{i_2}(t)\| \leq \cdots \leq \|p_{i_{|\mathcal{N}'_k(t)|}}(t)\|$. For each $\iota = 1, \dots, |\mathcal{N}'_k(t)|$, we sequentially join the index i_ι to the set $\mathcal{N}'_{k,\mathrm{occ}}(t)$ if and only if $|\angle(p_\iota - q_k) - \angle(p_\phi - q_k)| > \theta$ for all $\phi \in \mathcal{N}'_{k,\mathrm{occ}}(t)$. We use the parameter $\theta = \pi/36$.

2.3.2 Online-Target Switching

The Online-Target Switching (OTS) algorithm proposed by [34] is applied by judging the swarm separation. We implement this algorithm by replacing $\xi_k(t)$ in

(6) with $\xi_k^{\rm ots}(t)$ defined by

$$\xi_k^{\rm ots}(t) = \begin{cases} \bar{p}_k(t) + d^{\rm ots}\,\phi\big(\bar{p}_k(t) - x_g\big), & \text{if } \|p_k^{\#}(t) - \bar{p}_k(t)\| \le R^{\rm ots}, \\ p_k^{\#}(t) + d^{\rm ots}\,\phi\big(p_k^{\#}(t) - \bar{p}_k(t)\big), & \text{otherwise} \end{cases} \tag{10}$$

where

$$\bar{p}_k(t) = N_k(t)^{-1} \sum_{i=1}^{N_k(t)} p_i(t) \tag{11}$$

is the mass center of the swarm that the steering agent k can observe,

$$p_k^{\#}(t) = \operatorname*{arg\,max}_{p \in \{p_i(t)\}_{i \in [N_k(t)]}} \|\bar{p}_k(t) - p\| \tag{12}$$

represents the position of the passive agent farthest from the mass center, and $R^{\rm ots} = r^{\rm ots}\sqrt{N_k(t)}$ determines the size of the radius based on the number of the sheep, which is the same setup as the original algorithm. We choose $r^{\rm ots} = 10$ and $d^{\rm ots} = 25$ so that there is an appropriate distance of $R^{\rm ots}$ and $d^{\rm ots}$ between the steering agent and the swarm. In this way, the steering agent can maintain the swarm shape when changing the target position $\xi_k^{\rm ots}(t)$ in Equation (10).

2.4 Centralized Algorithms for Multiple Steering Agents

In this section, we investigate the role of coordination in centralized multi-agent shepherding algorithms, with a focus on its function during shepherding. The methods described in Section 2.4 are specifically designed for scenarios with multiple shepherds and rely on inter-shepherd communication to form coordinated formations. However, such coordination can sometimes restrict the individual flexibility of agents, potentially reducing overall performance. By comparing these methods with the proposed algorithm, which operates without communication, we aim to provide a clearer rationale for the necessity of communication in shepherding tasks.

2.4.1 Point-offset Circling Control

The shepherding algorithm proposed by [27] proposes a shepherding algorithm in which steering agents form an arc formation to steer the swarm. Within the algorithm, at each time a central controller computes the center of the mass $\bar{p}(t)$ and the position $p^{\#}(t)$ of the passive agent farthest from the center similarly in (11). Then, the controller regards the swarm as a circle with center $\bar{p}(t)$ and radius $R_s(t) = \|p^{\#}(t) - \bar{p}(t)\|$. The controller then generates a circle having the same center but with a larger radius, and directs each steering agent to move toward a point on the larger circle. Each steering agent is placed evenly on the larger circle.

Specifically, within the algorithm, the controller computes the following quantities at each time t:

$$\begin{aligned} R^{\text{circle}}(t) &= \alpha^{\text{circle}} R_s(t), \\ \Delta_k^{\text{circle}} &= \Delta \frac{(2k - m - 1)}{(2m - 2)}, \\ \theta(t) &= \angle(\bar{p}(t) - x_g) \end{aligned}$$

where $R^{\text{circle}}(t)$ represents the radius of the larger circle whose radius is controlled by parameter $\alpha^{\text{circle}} \geq 1$, Δ_k represents the degree of kth steering agent for determining its placement on the circle, and $\theta_{\bar{p}}(t)$ represents the angle of the center of the swarm in the counterclockwise direction with respect to the positive direction of x-axis. Each steering agent needs to know its index k within the total number M of steering agents. The controller then directs each steering agent to move toward its target position defined by

$$\xi_k^{\text{circle}}(t) = \bar{p}(t) + R^{\text{circle}}(t) \begin{bmatrix} \cos \Delta_k^{\text{circle}} + \theta(t) \\ \sin \Delta_k^{\text{circle}} + \theta(t) \end{bmatrix}.$$

Within our simulation, we use $\alpha^{\text{circle}} = 1.5$ and $\Delta = 2\pi/3$.

2.4.2 Potential-based Caging

The shepherding algorithm proposed by [31] employs a caging formalism in robotic manipulation and guides a group of passive agents to the goal region safely and with provable guarantees. The cage is constructed by a regular n-sided polygon and has the steering agents as its vertices. The distance between the center of the swarm and the vertex is set as R^{cage} determined by

$$\begin{aligned} R^{\text{cage}}(t) - R_s(t) - d_{\text{CSM}} &= R^{\text{cage}}(t)\sin(\pi/n), \\ \Delta_k^{\text{cage}} &= \frac{2k}{m}\pi, \end{aligned}$$

where d_{CSM} is the minimal required distance between the passive agents and the point. Then, the target position for the kth steering agent is set as

$$\xi_k^{\text{cage}}(t) = \bar{p}(t) + R^{\text{cage}}(t)\begin{bmatrix}\cos\Delta_k^{\text{cage}} \\ \sin\Delta_k^{\text{cage}}\end{bmatrix} + \alpha^{\text{cage}}\phi(\bar{p}(t) - x_g).$$

We remark that we are introducing the term parameter $\alpha^{\text{cage}}\phi(\bar{p}(t) - x_g)$ so that the algorithm can achieve guidance of the swarm into the goal region. In the caging process, each steering agent moves to a vertex close to itself as its target position while making sure that no vertex is shared with multiple steering agents. We use $d_{CSM} = 0.05R_s(0)$ and $\alpha^{\text{cage}} = -8$.

2.5 Experiments

In this section, we present numerical simulations to evaluate the performance of the proposed algorithm, which operates without communication between steering agents. These simulations aim to investigate how effective shepherding can be achieved without relying on communication. To validate this, we compare the proposed approach with other methods, including baseline algorithms that increase the number of agents without communication and centralized methods that incorporate communication while employing a formation strategy. This comparison helps isolate the effects of communication from other factors, providing a clearer understanding of its role and the necessity of its inclusion in multi-agent systems.

For each of the initial placements, all trials are to be terminated when all the passive agents are within the goal region, or after 3000 steps regardless of the result of navigation. In the former case, we label the trial for the initial configuration as a success.

2.5.1 Parameter Values

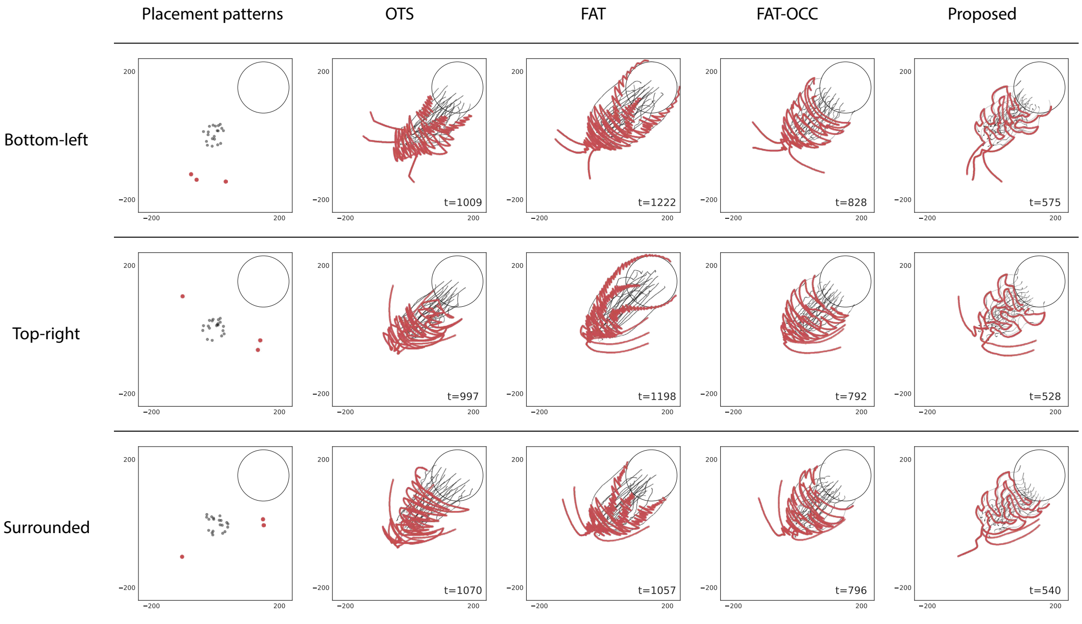
We assume that there exist N passive agents to be guided in a two-dimensional space. We suppose that, at the initial time, these agents are placed uniformly and randomly within a disc centered at the origin, with an initial radius $R_s(0)$. Here, the origin represents the center of the initial agent placement distribution. We design the pattern of the initial distribution as 1) a small swarm: $N = 20, R_s(0) = 40$, 2) a large swarm: $N = 50, R_s(0) = 60$, and 3) two separate swarms: $N = 20, R_s(0) = 40$ for one swarm and $N = 30, R_s(0) = 50$ for another swarm. The parameters of the passive agent model are set as

$$c_1 = 200, c_2 = 0.2, c_3 = 0.02, c_4 = 400, c_5 = 0.1 \tag{13}$$

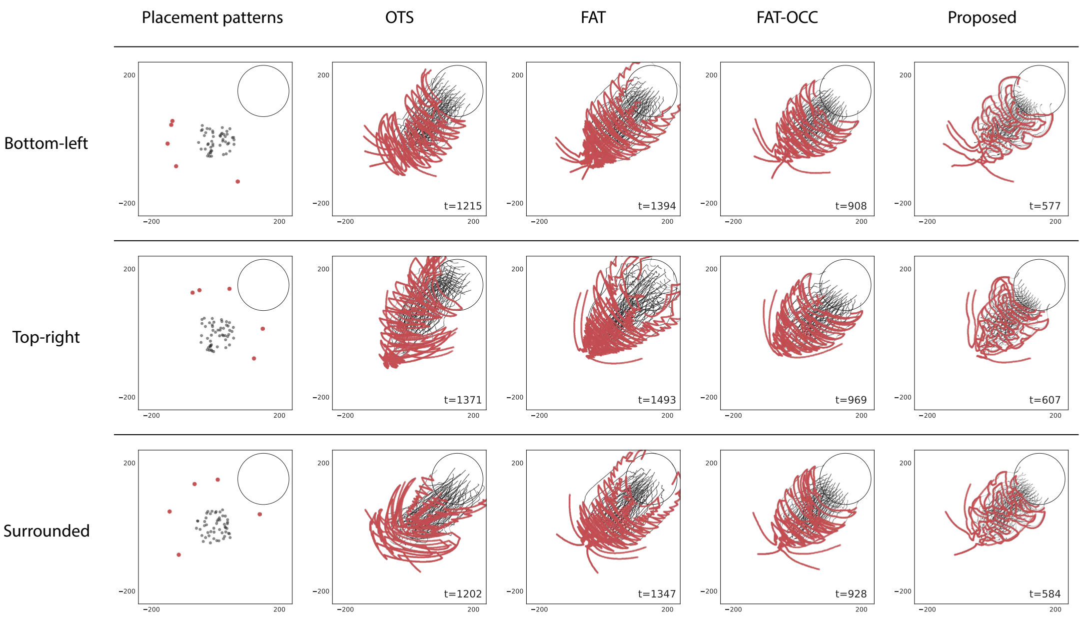
and $r = 50$. We call this parameter set or scenario as *default*. The goal G is supposed to have the center $x^g = [150, 150]^\top$ and radius $R^g = 80$. The radius size R_g is reasonably determined based on the number of passive agents in the simulation. For the comprehensiveness of our experiment, we prepare the following three different placement patterns of the steering agents; steering agents are initially 1) placed around at the bottom-left of the passive agents (*bottom-left*), 2) placed around at the top-right of the passive agents (*top-right*), and 3) surrounding the passive agents (*surrounding*). For each of the placement patterns, we randomly generate 100 trials for different initial placements of agents. Samples of the initial placements are shown in the first column of Figure 6.

2.5.2 Experiment Results

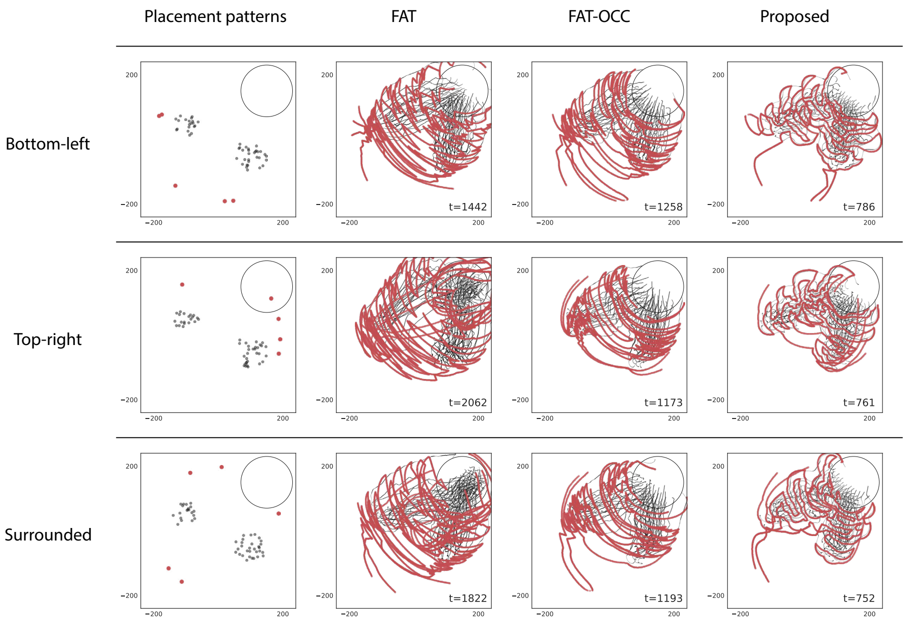
We conduct simulations to illustrate the effectiveness of the proposed algorithm. Within the simulations, we conduct shepherding of a swarm following one of the three initial distributions of passive agents and three placements of steering agents using one of the four algorithms. We illustrate the performance of the algorithms using the trajectories of agents. Toward this end, for each pair of the four algorithms and three placement patterns, we pick the quickest trial among 100 initial placements. The trajectories and their corresponding completion time are shown in Figure 6. We can observe that the trajectories of the steering agents in the proposed algorithm are smoother than those of the three baseline algorithms, confirming the effectiveness of the decentralized mechanism of the proposed algorithm. For guiding the two separate swarms, we find through numerical simulations that the switching algorithm is not capable of performing the shepherding task, so we only compare the three remaining algorithms in Figure 6c. We also observe the FAT algorithm tends to consume more time or fail depending on the cases of the initial placements. After examining the simulation data, we identify the following problems with the FAT algorithm. One problem is that guiding a large swarm can consume excessive time due to long traversal distances. Another problem is that a swarm tends to be scattered when the steering agent chases the passive agent on the opposite side of the swarm. The scattering makes the shepherding process more difficult and increases consumed time.



(a) Guiding a small swarm in the default passive agent model with $M = 3$.



(b) Guiding a large swarm in the default passive agent model with $M = 5$.

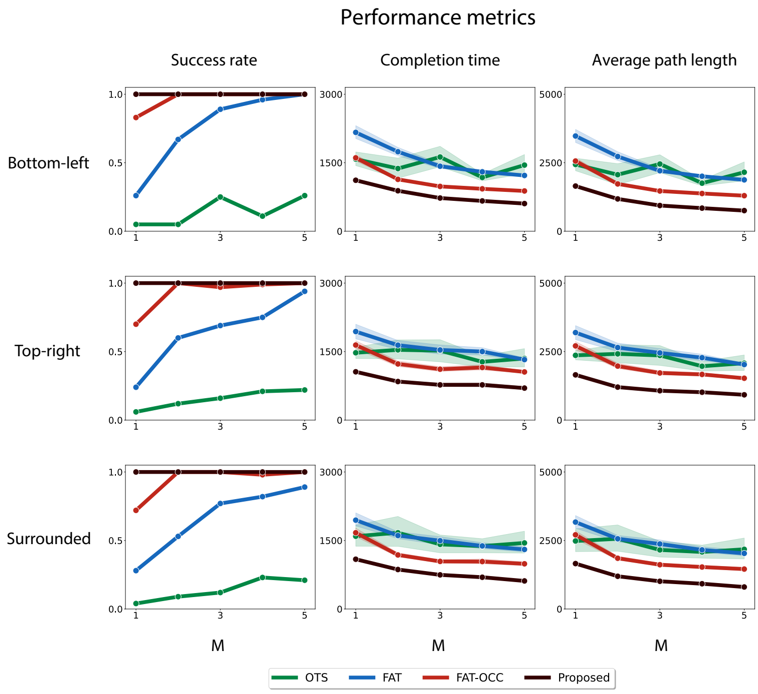


(c) Guiding two separate swarms in the default passive agent model with $M = 5$. The trajectory of the online-target switching (OTS) algorithm is not shown because of its failure.

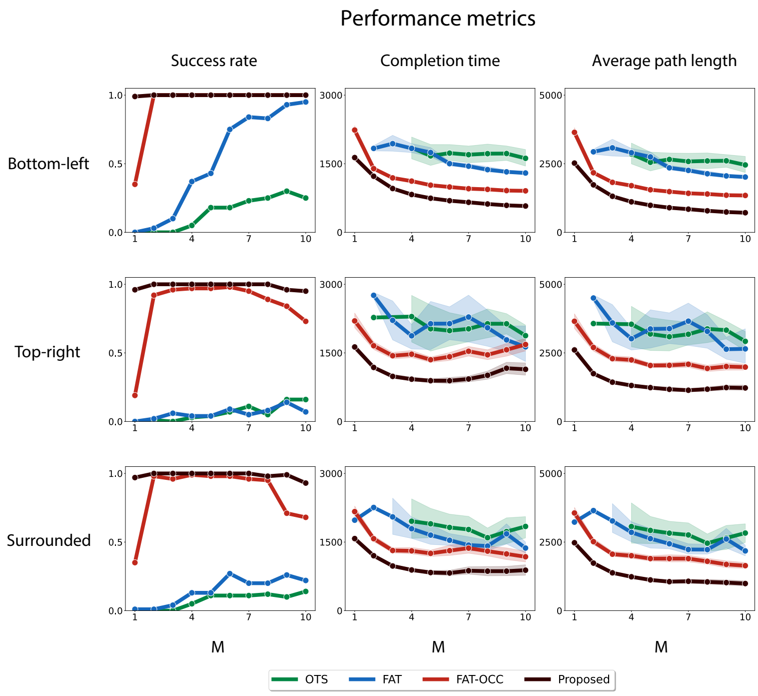
Figure 6. Initial placements and corresponding trajectories for guiding three types of swarms in the default passive agent model. 1st column: Samples of initial placements. 2nd to the last columns: Trajectories of the quickest navigations among those performed for randomly generated 100 initial placements. Circle: goal region. Red dots: steering agents. Gray dots: passive agents. The numbers at the bottom-right indicate the time at which the shepherding navigation is completed. It is remarked that the initial placements in each row are not necessarily the same.

For further evaluation and comparison of the proposed and the baseline algorithms, we introduce the following three performance measures. First, the success rate of an algorithm for a placement pattern is defined as the rate of successful trials among randomly generated 100 initial placements. Second, we define the completion time as the execution time of the algorithm in its successful trials. Finally, the average path length is defined as the average of the mean traveling distance $M^{-1}\sum_{k=1}^{M}\sum_{t}\|v_k(t)\|$ of steering agents in successful trials.

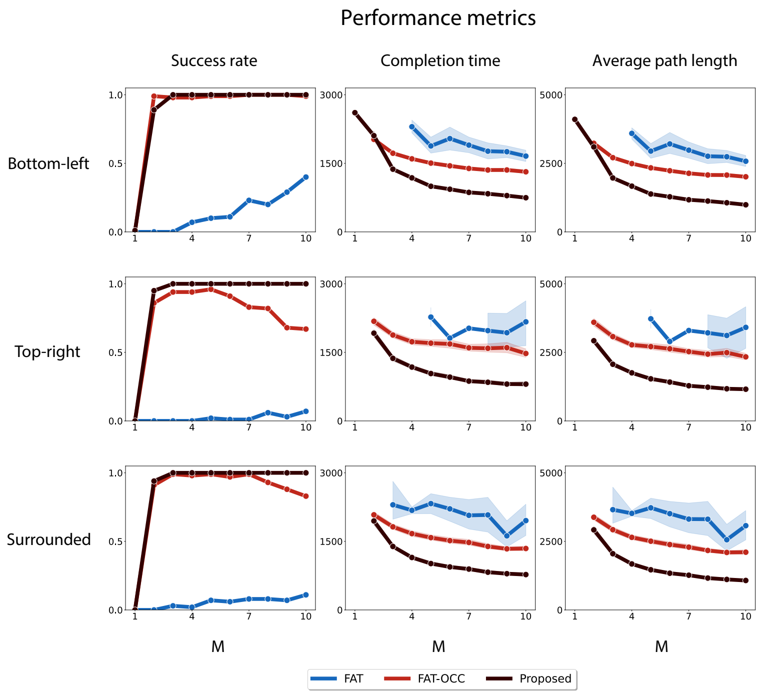
Figure 7 represents how these three performance measures depend on the number of steering agents for each of the algorithms. We observe that the proposed algorithm achieves almost 100% success rate regardless of the number of steering agents and placement patterns, which confirms the effectiveness and scalability of the proposed algorithm. We can also observe that the proposed algorithm outperforms the baseline algorithms in completion time and average path length. Furthermore, the average completion time and average path length steadily decrease with respect to the number of steering agents. These trends suggest that the proposed algorithm allows stable and synergistic coordination of steering agents for the navigation of passive agents.



(a) Guiding a small swarm in the default passive agent model. Horizontal axes represent the number of steering agents M from 1 to 5.



(b) Guiding a large swarm in the default passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.



(c) Guiding two separate swarms in the default passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.

Figure 7. Performance of the algorithms for guiding three types of swarms in the default passive agent model. Horizontal axes represent the number of steering agents. 1st column: the rate of successful navigation. 2nd column: success time. 3rd column: average traversal distance of steering agents. In the 2nd and 3rd columns, a solid line draws an estimate of the mean value and shaded areas describe the confidence interval for that estimate.

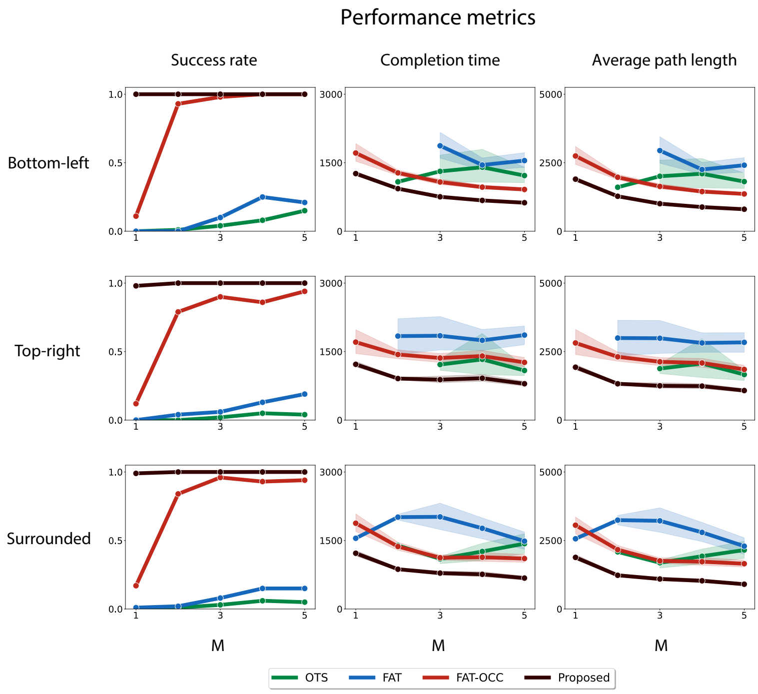
For a more thorough comparison, let us consider two other scenarios in which the parameters of the passive agents are different from the ones used in previous simulations. In the first scenario, we consider the parameters

$$c_1 = 250, c_2 = 0.2, c_3 = 0.025, c_4 = 500, c_5 = 0.1.$$

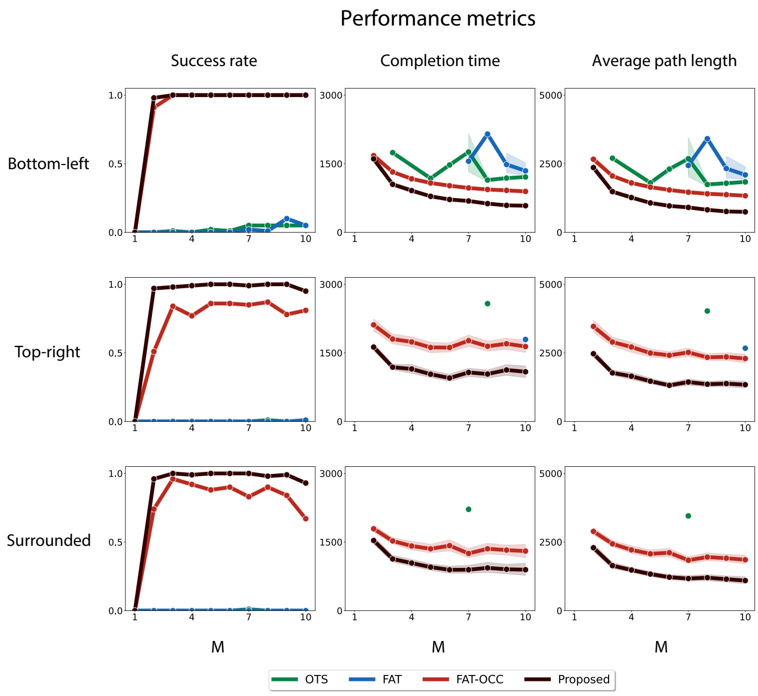
These values are all no less than the corresponding ones of the default scenario in (13). For this reason, we expect that the swarm with these parameters is more sensitive to the movement of the steering agents. Let us call this scenario *sensitive*. On the other hand, we prepare the other additional scenario to perform comparisons for the swarm that is harder to navigate. For this reason, in the second scenario, we use the parameters

$$c_1 = 150, c_2 = 0.2, c_3 = 0.015, c_4 = 300, c_5 = 0.1.$$

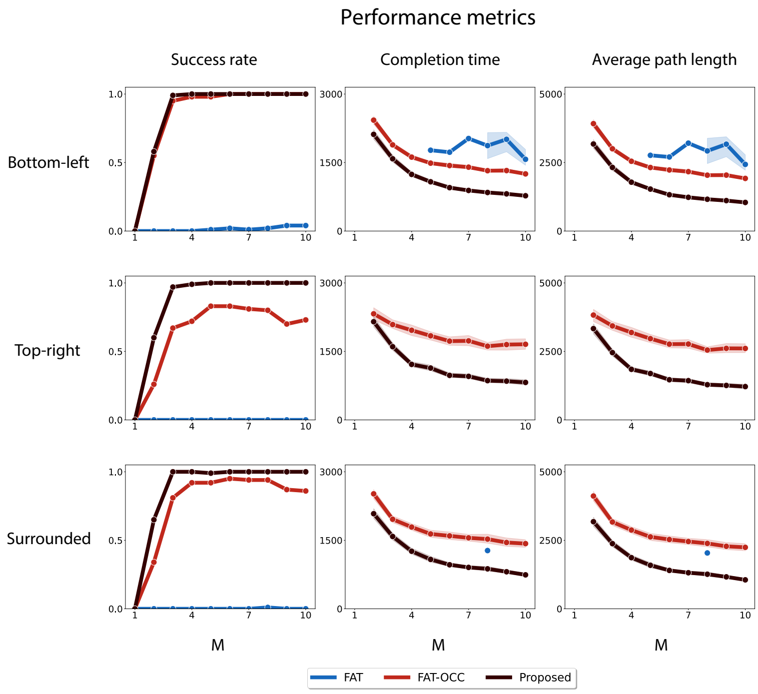
Because these values are all no greater than the corresponding ones of default, we call this scenario *insensitive*. Now, under these two additional scenarios, we conduct the same set of simulations that we did for the default scenario (13). The results of the simulations in the scenarios sensitive and insensitive are illustrated in Figure 8 and Figure 9, respectively. We can confirm that the proposed algorithm always shows higher success rates as well as lower completion time and shorter path lengths. In Figures 8b and 9b, we also observe a slight decrease in the success rate of the proposed algorithm. After investigating the failure cases, we find the following reasons. One reason is the interference behaviors among multiple steering agents when steering agents may coincidentally choose the same target and drive the target further away from the goal without returning. Another reason is due to the behaviors of the passive agents; when the parameters of the passive agent model are changed to be more sensitive, it can be difficult for the steering agents to include all the passive agents inside the goal region simultaneously.



(a) Guiding a small swarm in the sensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 5.

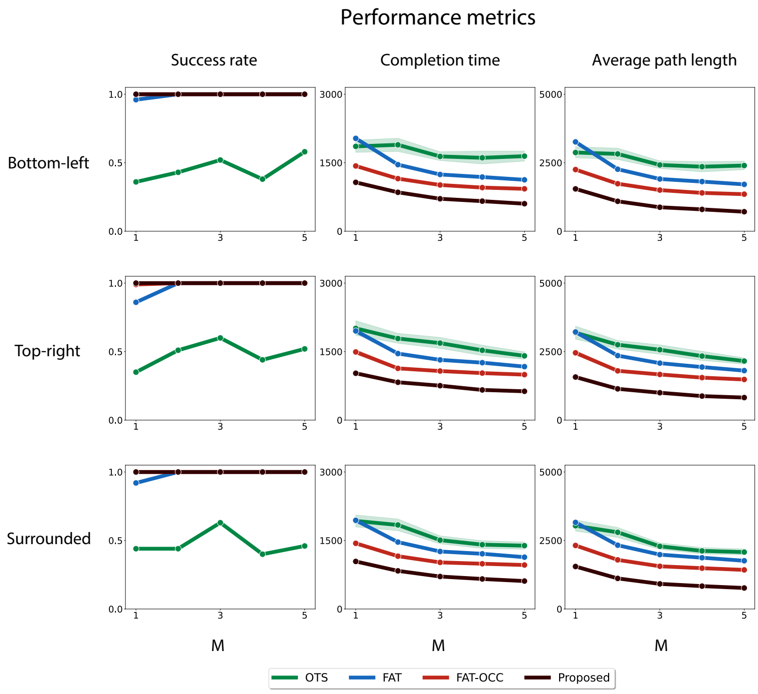


(b) Guiding a large swarm in the sensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.

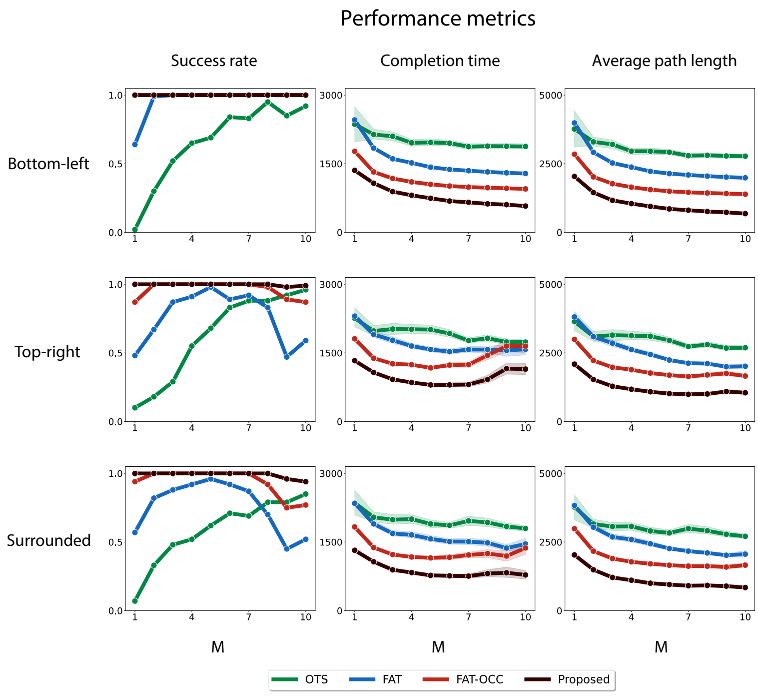


(c) Guiding two separate swarms in the sensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.

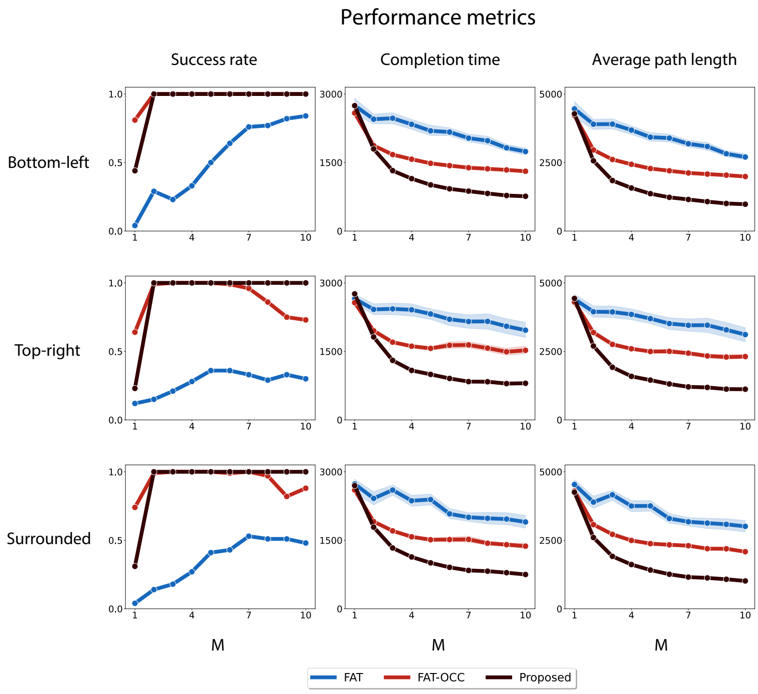
Figure 8. Performance of the algorithms for guiding three types of swarms in the sensitive passive agent model. Horizontal axes represent the number of steering agents.



(a) Guiding a small swarm in the insensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 5.



(b) Guiding a large swarm in the insensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.



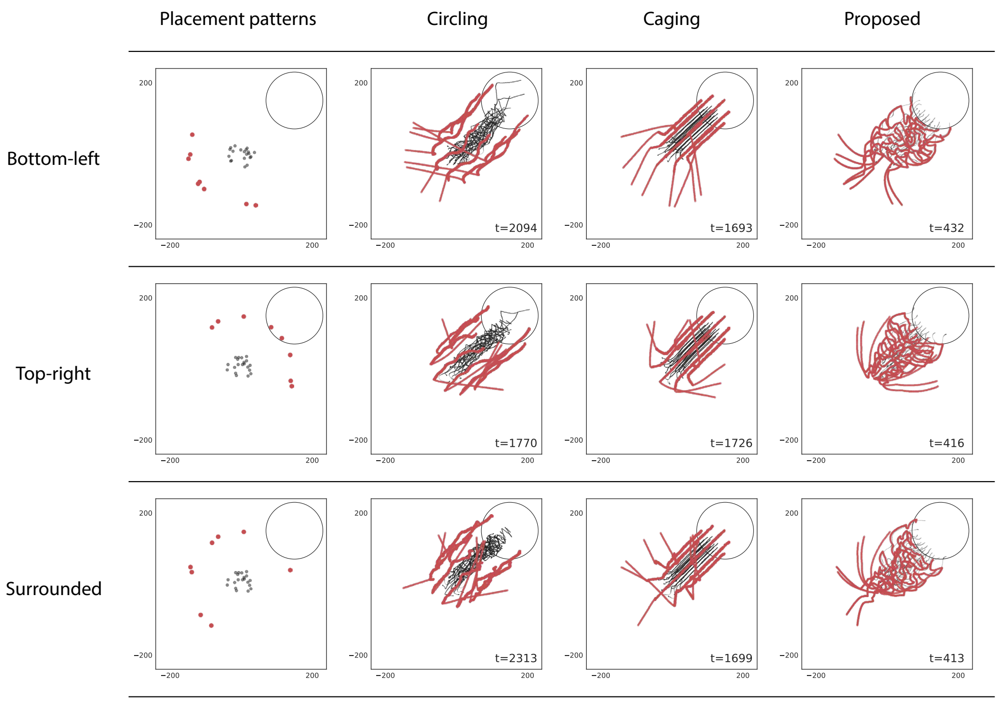
(c) Guiding two separate swarms in the insensitive passive agent model. Horizontal axes represent the number of steering agents M from 1 to 10.

Figure 9. Performance of the algorithms for guiding three types of swarms in the insensitive passive agent model. Horizontal axes represent the number of steering agents.

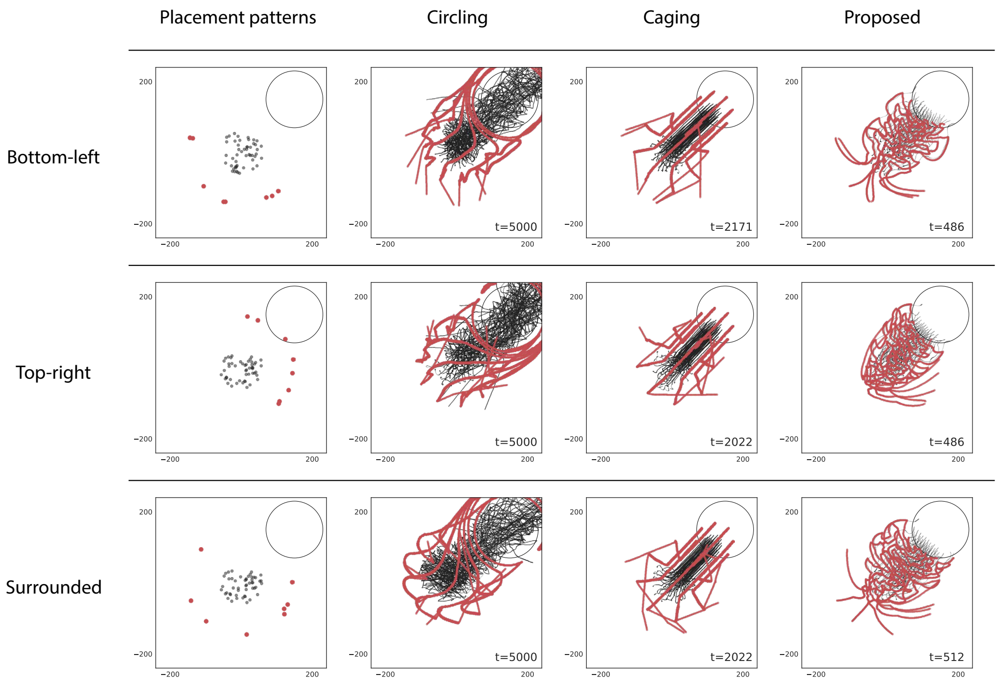
We then conduct simulations to compare the proposed algorithm with both centralized shepherding algorithms. In the centralized algorithms, each steering agent moves strictly according to the design, with the dynamics of the steering agent presented in Equation (9) being modified by setting $d_2 = 0$, $d_3 = 0$, and $d_4 = 0$. From our preliminary simulations, we found that the algorithms presented above do not perform well in some situations. Therefore, in our simulations, to make the coordination of multiple steering agents stable, we modify the radius $R_s(t) = \min\{\|\bar{p}(t) - p^{\#}(t)\|, \beta\|\bar{p}(0) - p^{\#}(0)\|\}$ to prevent failure when the swarm is dispersed during shepherding and we choose $\beta = 1.25$. Further, we define that the algorithm for point-offset circling control takes the same strategy to allocate the steering agents to their target positions. The maximum time step is set to 5000 to ensure the completion of the shepherding.

The trajectories and the performance for guiding the small and large swarms are shown in Figures 10a and 11a and Figures 10b and 11b, respectively. Simulation results indicate that for these two algorithms, the average completion time increases and the success rate decreases as M increases. On the other hand, when guiding a swarm with large N and $R_s(0)$, the success rate is not necessarily high. We analyze this poor performance due to the large interaction distances between the steering agents and the swarm. For the case of two separate swarms, we choose not to present the simulation results because we found through numerical simulations that these two centralized algorithms are incapable of performing the shepherding task when multiple separate swarms exist. From the simulation results based on the two other sets of the passive agent model, in Figures 12a and 12b and Figures 13a and 13b, we observe that the performance of these two algorithms is greatly influenced by the parameter setting of the passive agent model. After examining the simulation data, we find that although the poor performance of centralized algorithms is counterintuitive, these algorithms, which rely on multiple steering agents moving in a fixed shape or formation to guide swarms, are neither effective nor robust.

Based on the results above, we have observed that the advantage of the proposed algorithm is that, although no communication is used, better performance is achieved by assigning the movement of each steering agent to different target passive agents. Specifically, the algorithm leverages cooperation among multiple agents using observable position information, even though the agents do not communicate with each other to share additional information. This approach improves scalability to changes in initial placements, accommodates increases in shepherd numbers, and enhances robustness to reductions in sensing accuracy.

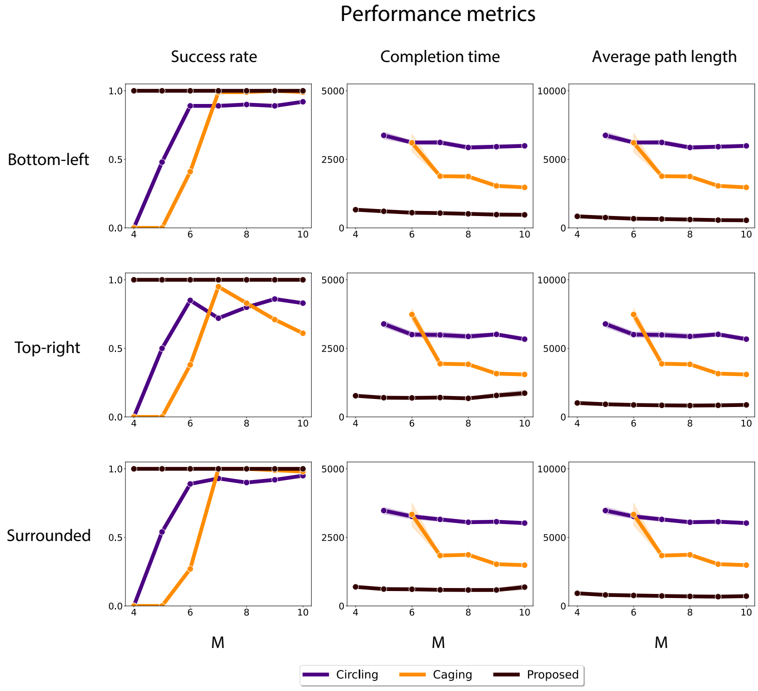


(a) Initial placements and corresponding trajectories for guiding a small swarm.

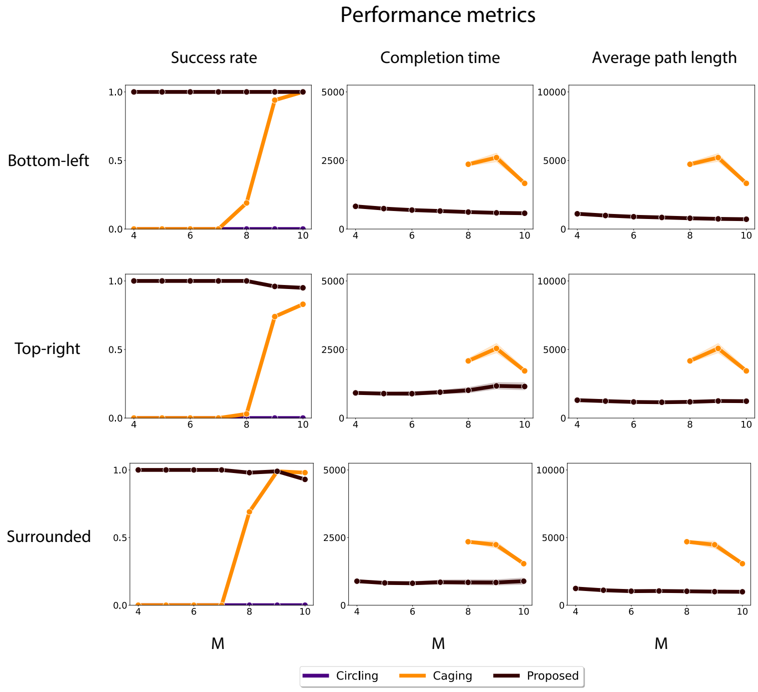


(b) Initial placements and corresponding trajectories for guiding a large swarm. The circling algorithm of shepherding continues until the maximum time step.

Figure 10. Initial placements and trajectories for guiding small and large swarms in the default passive agent model, compared with the proposed algorithm and centralized shepherding algorithms. The number of steering agents is set to $M = 8$.

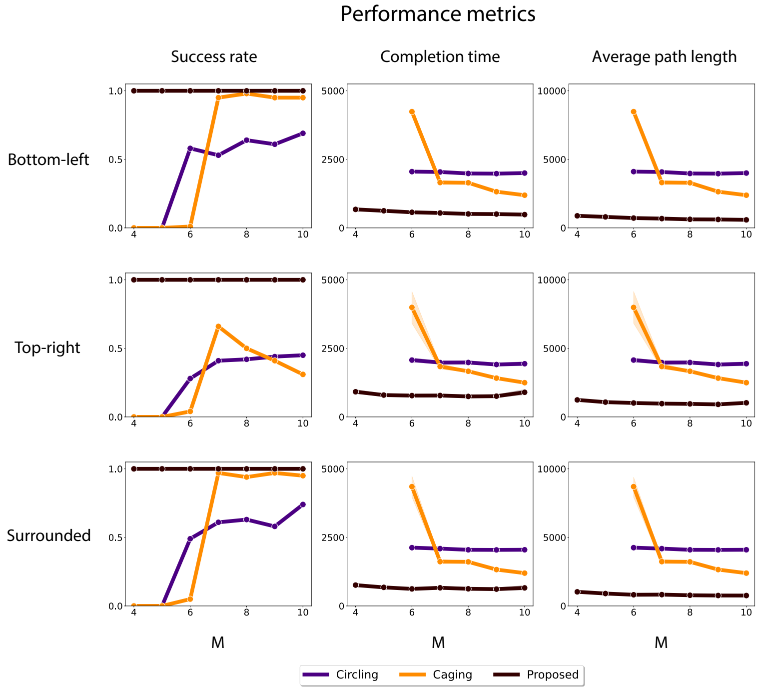


(a) Guiding a small swarm in the default passive agent model.

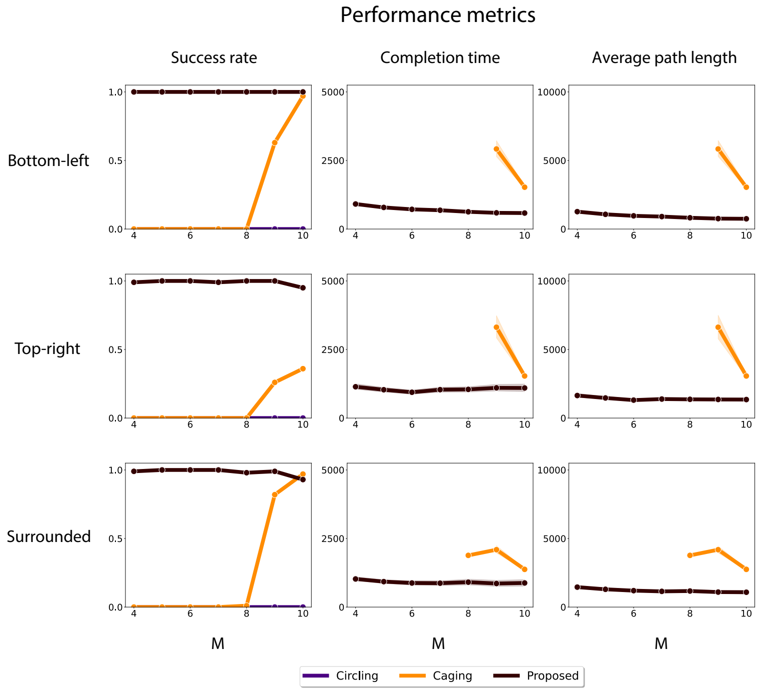


(b) Guiding a large swarm in the default passive agent model.

Figure 11. Performance comparison between the centralized algorithms and the proposed algorithm for guiding a small and large swarm in the default passive agent model. Horizontal axes represent the number of steering agents M from 4 to 10. The y-axis indicates completion time and average path length.

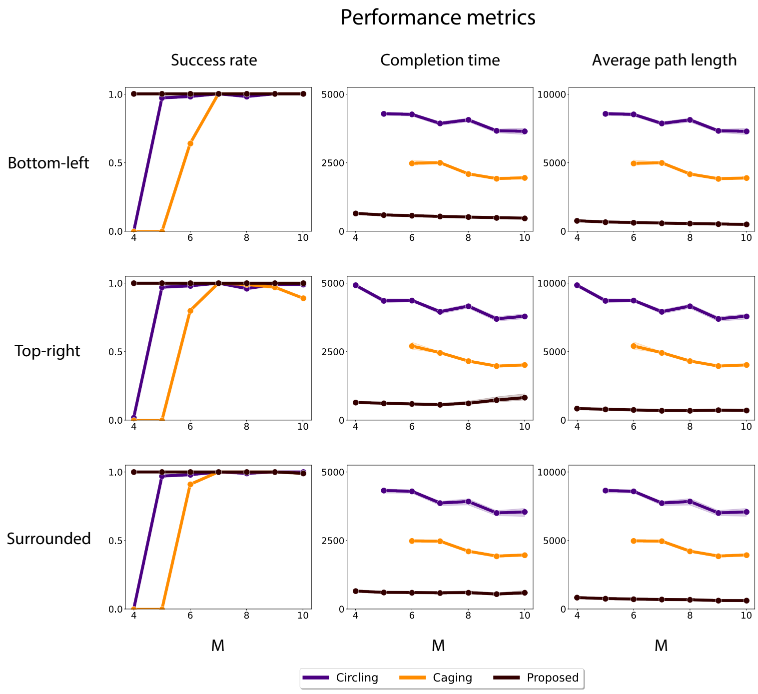


(a) Guiding a small swarm in the sensitive passive agent model.

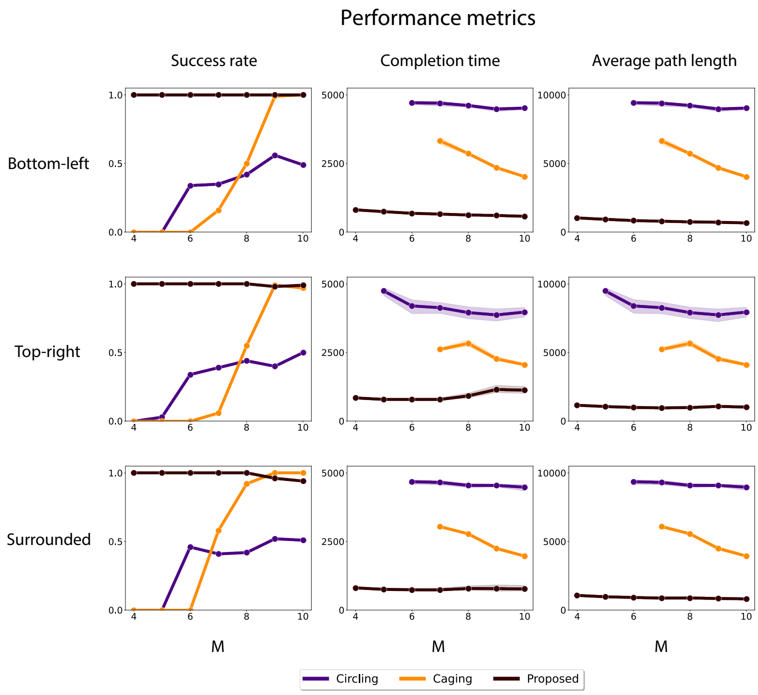


(b) Guiding a large swarm in the sensitive passive agent model.

Figure 12. Performance comparison between the centralized algorithms and the proposed algorithm for guiding a small and large swarm in the sensitive passive agent model. Horizontal axes represent the number of steering agents M from 4 to 10. The y-axis indicates completion time and average path length.



(a) Guiding a small swarm in the insensitive passive agent model.



(b) Guiding a large swarm in the insensitive passive agent model.

Figure 13. Performance comparison between the centralized algorithms and the proposed algorithm in the insensitive passive agent model. Horizontal axes represent the number of steering agents M from 4 to 10. The y-axis indicates completion time and average path length.

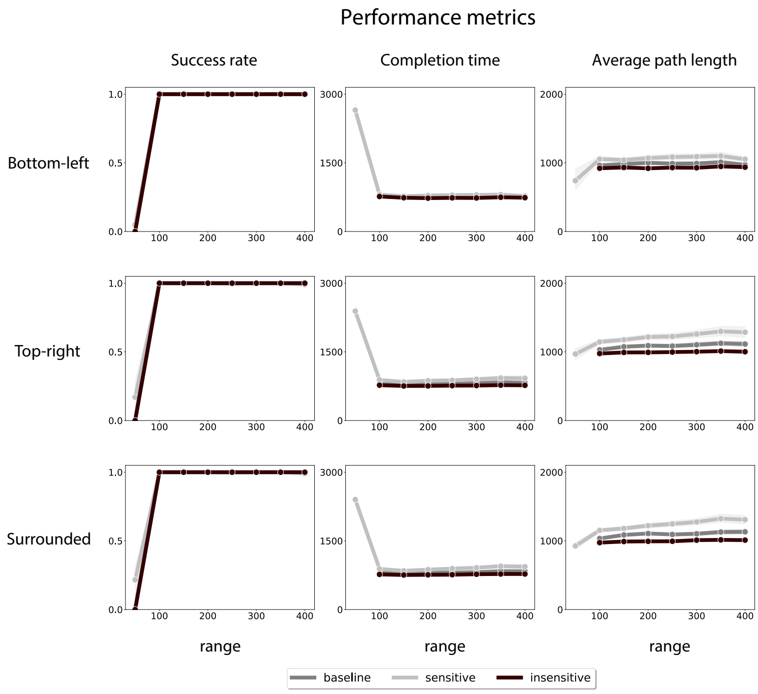
2.5.3 Experiments with Further Reduced Information

In this subsection, we numerically evaluate the robustness of steering agents under further reduced sensing accuracies. In this context, we define robustness as the ability to achieve success and maintain effectiveness despite reductions in sensing range and accuracy. Throughout this subsection, we use 2) the large swarm to be the same set of initial placements as the ones we used in the last subsection. The passive agents were modeled as default, sensitive, or insensitive. The number of the steering agents is fixed as $M = 5$. Because our main objective in this subsection is to investigate the robustness properties of the proposed algorithm, we do not conduct simulations of existing methods.

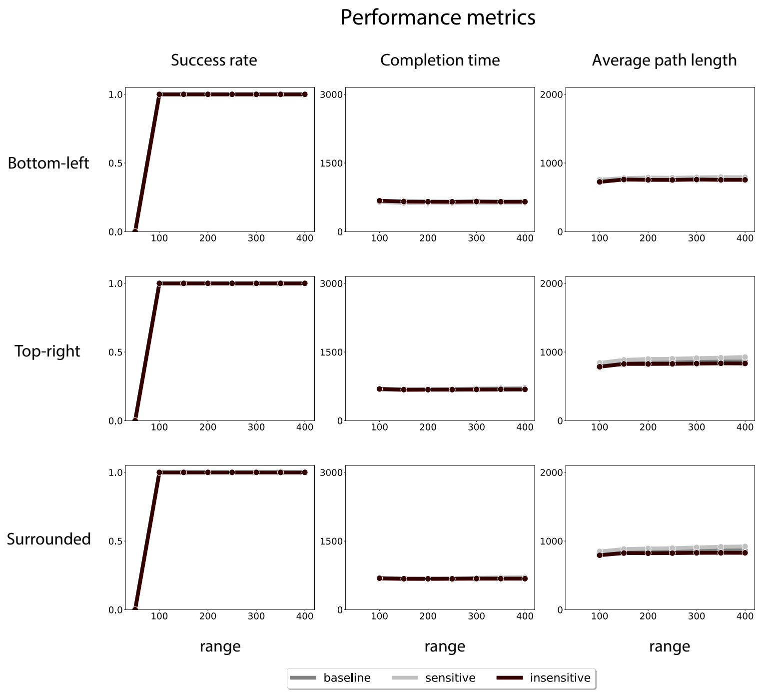
We first examine the performance of the proposed algorithm with respect to the change in the sensing range of the steering agents and the passive agents. As for the sensing range of the shepherd, we change r' from its default value 300 and vary within the set $\{50, 100, 150, ..., 400\}$. Also, we prepare two scenarios on the sensing range r of the passive agents; $r = 50$ and $r = 100$. We present how the success rates, completion times, and average path lengths depend on r' in Figure 14a ($r = 50$) and Figure 14b ($r = 100$). According to the results, different sizes of the sensing ranges r of passive agents cause changes in the swarm behaviors to influence the shepherding performance. For these two values of r, the success rate of shepherding drops when the sensing range r' of the steering agent is short. This observation suggests that, for the proposed algorithm to be effective, we should avoid employing a steering agent having a too short sensing range.

We then evaluate the performance of the proposed algorithm under sensing errors of the steering agents. In this simulation, we assume that the sensing of the steering agent to the positions of other agents and the goal is subject to additive noise in the form of $d_5\sigma(t)$ where d_5 is a positive weight, and the random vector $\sigma(t)$ is generated in the same way as the random vector $u_{i5}(t)$ in Equation (2). Importantly, we allow each agent to continue targeting the correct passive agent for guidance according to

Equation (6), unaffected by sensing errors. This means that while the target agent i is selected correctly, the position $p_i(t)$ is perceived with error as $p_i(t) + d_5\sigma(t)$, as described in Equation (3), which leads to movement deviations for the steering agents. In Figure 15, we show how the performance of the proposed algorithm depends on the weight d_5. We confirm that the proposed algorithm tolerates relatively sensing error increased to $d_5 = 10$ in any of the initial placements. This observation indicates that the strategy for selecting targets for each agent is crucial, and the proposed algorithm remains robust to sensing errors as long as the strategy is implemented correctly.



(a) The sensing range of the passive agent is set as the default value $r = 50$.



(b) The sensing range of the passive agent is enlarged as $r = 100$.

Figure 14. Performance of the proposed algorithm for guiding swarms in different sensing ranges for steering agents. Horizontal axes represent the sensing range of steering agents r'. The sensing range of the passive agents is varied between $r = 50$ and $r = 100$.

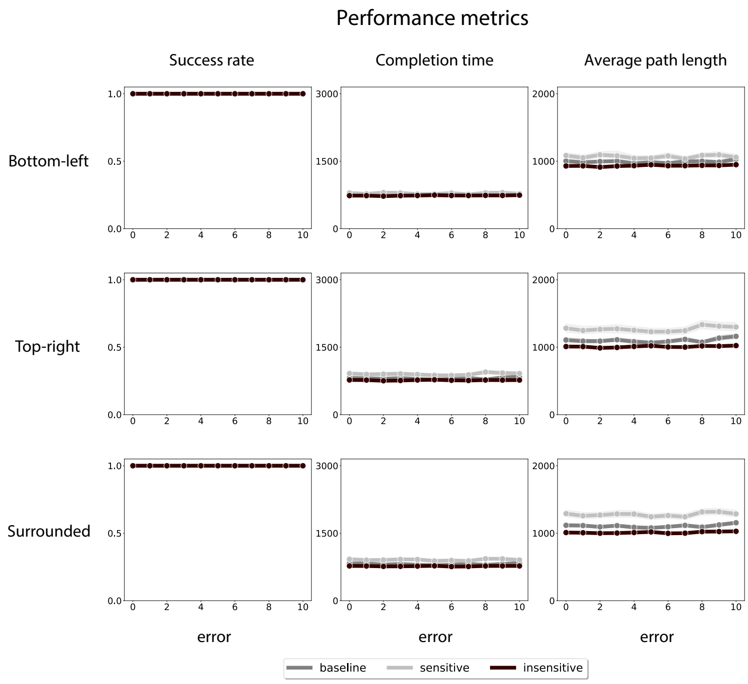


Figure 15. Performance of the proposed algorithm for guiding three sets of swarm in considering sensing error for steering agents. Horizontal axes represent the value d_5 of the sensing error for steering agents.

2.6 Summary

In this chapter, we studied the shepherding problem with multiple steering agents unable to communicate with each other. The results demonstrate that communication between steering agents is not required for successful shepherding when cooperation relies on relative situational sensing, even in the presence of sensing errors. Specifically, we have first presented a model of passive agents in the presence of multiple steering agents. We have then proposed a distributed and communication-free algorithm with multiple steering agents to aggregate the passive agents by location-based self-planning. We have also compared the proposed algorithm with other default algorithms with and without centralized coordination. Finally, we have confirmed the robustness of the proposed algorithm via extensive numerical simulations in various situations, including different levels of sensing errors and mobility-related accidents.

The natural cooperative strategy for the steering agents adopted by the proposed algorithm proves to be successful, effective, and robust compared to traditional algorithms. In our simulations, the parameter value α used to balance factors when selecting the target agent for each steering agent in Equation (6) was fixed at a specific value. However, performance could potentially be improved by exploring alternative formulations of the equation and assigning differentiated values to α to enhance cooperation. Moreover, while the algorithm operates without communication, incorporating communication or combining the strategy with other approaches could increase flexibility and expand the range of applications.

In future research, we plan to investigate whether the proposed communication-free coordination mechanism can be extended to other types of navigation tasks. Moreover, we have observed that while we model passive agent movement based on pre-assumed models and parameters, we observed that if passive agents deviate from these settings, irregular spacing and unexpected reactions to steering agents may occur. Therefore, one of the future works is to validate the shepherding algorithms using different models imitating the practical behaviors of organisms or robots.

Chapter 3

Shepherding Control by Bearing-only Algorithm

3.1 Introduction

Our second study focuses on further reducing the required sensing information and developing an algorithm for successful shepherding under these constraints. Previous studies on the shepherding problem have predominantly assumed that steering agents have sufficient sensing capabilities [38], i.e. each agent can recognize passive agents by their positions and velocities, in conjunction with the positions and velocities of other steering agents, and the position of the goal. However, in practice, robots performing a guiding task may not be able to collect all the expected sensing results from the surrounding environment [23]. Thus, the study of shepherding incorporating such limitations on the sensing capability of steering agents has been conducted from various perspectives, such as local-camera-based observation [37], lack of computation ability or memory [45], and lack of coordination among multiple steering agents [42].

On the other hand, in the context of relative bearing measurements, this term refers to determining the relative direction or angle of a target object with respect to a reference point or axis. Each agent can measure only the relative bearings of its neighboring agents, without access to their relative distances or proximities [46]. The process of controlling such agents to achieve desired formation patterns is referred to as bearing-only formation control, focusing on accurately coordinating all moving agents [47, 48]. However, using one set of agents to guide another set of unmanoeuvrable agents solely based on bearing measurements remains a challenging problem in swarm control.

This study presents a bearing-only algorithm for shepherding with limited information. Leveraging the proposed algorithm, we find the essential sensing information by the algorithm to guarantee the success of shepherding. The algorithm is inspired by the strategy of a two-stage approach, which divides the movements of a steering agent by initially orienting its position relative to its target swarm and then driving the swarm towards the goal [34, 39]. The target swarm is one of the swarms that is selected by the steering agent among multiple swarms as the target for chasing. Specifically, we first introduce an algorithm using a steering agent, design strategies to allow multiple steering agents to cooperate through reduced collisions and improved efficiency by sharing limited knowledge of bearing measurements (i.e., direction from each position to the estimated center of each target swarm), and then apply distributed strategies for steering agents to guide multiple swarms. The experiments are conducted for various initial placements with different parameter values for the passive agents, to evaluate the effectiveness and robustness of the proposed algorithm. Finally, we discuss the influence of bearing measurement accuracy and the role of communication between steering agents to understand the requirements of the proposed algorithm.

The remainder of this chapter is organized as follows. Section 3.2 outlines the knowledge of steering agents in bearing-only measurements. Section 3.3 presents a step-by-step description of the proposed algorithm. Section 3.4 shows experimental results to illustrate the functionalities and capabilities of the proposed algorithm under various configurations of parameter values and initial placements and presents an investigation of the essential amount of information.

3.2 Shepherding Knowledges

The problem of the shepherding task focuses on designing a shepherding algorithm for the steering agents to guide a set of passive agents into a designated goal region.

In this section, we describe the movements of each passive agent using an agent-based model to form swarm movements, present the goal setting, and design the knowledge of steering agents with bearing measurements.

We let a shepherd k guide a swarm using the following knowledge of bearing-only measurements. In the case of a single swarm, all passive agents are initially aggregated, and the relative distances between them are limited.

First, we assign each steering agent to observe the other agents under occlusion [37] with bearing measurements. Similar to how we defined Equation (1), we begin by constructing sets $\mathcal{N}'_k(t)$ and $\mathcal{M}'_k(t)$, and then define $\mathcal{O}'_k(t) = \mathcal{N}'_k(t) \cup \mathcal{M}'_k(t)$ to represent the set containing all other agents within the limited sensing range r'. For shepherd k, to get a subset $\mathcal{O}'_{k,\mathrm{occ}}(t)$, we initialise $\mathcal{O}'_{k,\mathrm{occ}}(t) = \emptyset$ and relabel the indices as $\mathcal{O}'_k(t) = \{x_1(t), \ldots, x_{|\mathcal{O}'_k(t)|}\}$ in such a way that $\|x_1(t) - q_k(t)\| \leq \|x_2(t) - q_k(t)\| \leq \cdots \leq \|x_{|\mathcal{O}'_k(t)|}(t) - q_k(t)\|$. For each $\iota = 1, \ldots, |\mathcal{O}'_k(t)|$, we sequentially join index ι to set $\mathcal{O}'_{k,\mathrm{occ}}(t)$ if and only if the angular difference from the other agent in $\mathcal{O}'_k(t)$ is larger than a constant θ_{occ}, which is $|\angle(x_\iota(t) - q_k(t), x_\upsilon(t) - q_k(t))| > \theta_{\mathrm{occ}}$ for any $\upsilon \in \mathcal{O}'_{k,\mathrm{occ}}(t)$. We then partition $\mathcal{O}'_{k,\mathrm{occ}}(t)$ in terms of passive agents and steering agents to update the sets $\mathcal{N}'_k(t)$ and $\mathcal{M}'_k(t)$, respectively.

For vectors x, y, and z, we define $\Theta_x(y, z) \in [-\pi, \pi)$ by

$$\Theta_x(y, z) = \angle(z - x, y - x)$$

to denote the angle between the vectors $z - x$ and $y - x$. In this study, the range of any angle is defined to be $[-\pi, \pi)$, wherein a negative value indicates clockwise rotation (right) and a positive value indicates counterclockwise rotation (left), to distinguish the right and left directions.

Then, from the position of shepherd k, the positions of the passive agents on the right and left sides of the swarm are given by

$$\begin{aligned} p_{k_r}(t) &= \arg\min_{p \in \{p_i(t)\}_{i \in \mathcal{N}'_k(t)}} \Theta_{q_k(t)}(g, p), \\ p_{k_l}(t) &= \arg\max_{p \in \{p_i(t)\}_{i \in \mathcal{N}'_k(t)}} \Theta_{q_k(t)}(g, p), \end{aligned} \tag{14}$$

while shepherd k knows only the direction from itself to these two positions. Figure 16 visualizes how the angles and positions are calculated. Subsequently, we assume that shepherd k has the following three vectors at time t:

$$Q_k(t) = \{\phi(p_{k_r}(t) - q_k(t)), \phi(p_{k_l}(t) - q_k(t)), \phi(g - q_k(t))\} \tag{15}$$

instead of the following positions: $\mathcal{N}'_k(t)$, $\mathcal{M}'_k(t)$, and x_g.

We then average the vectors to the left and right sides to obtain a vector to the estimated swarm center, denoted as $c_k(t)$, such that $\phi(c_k(t) - q_k(t)) = \phi(\phi(p_{k_r}(t) - q_k(t)) + \phi(p_{k_l}(t) - q_k(t)))$. The angle between the direction from shepherd k to estimated swarm centre $c_k(t)$ and the direction from shepherd k to goal x_g is then denoted as $\Theta_{q_k(t)}(c_k(t), x_g)$ based on knowledge $Q_k(t)$ defined in Equation (15). Additionally, we assume that shepherd k can memorize only $Q_k(t)$ at each time step and, therefore, cannot estimate the relative distance to any agent based on the change in angle over time. Because the steering agent cannot measure how far it has moved, we assign a fixed size for the velocity of each shepherd, denoted $\|v_k(t)\| = d$, where d is a positive constant.

3.3 Proposed Algorithm

The concept of our proposed algorithm is inspired by the online-target switching (OTS) algorithm [34], which demonstrates that observing and guiding the swarm as a single group can be effectively achieved using bearing measurements. We first introduce the proposed algorithm based on a single steering agent ($M = 1$) guiding a single swarm. Afterward, we extend the algorithm to allow for multiple steering

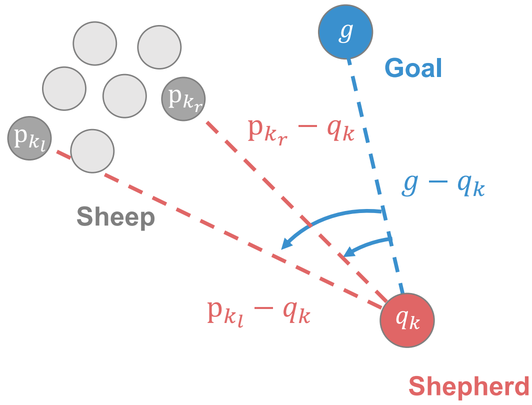


Figure 16. Illustration depicting how shepherd k observes the directions to the left and right positions (dark grey dots) of the swarm boundary relative to the goal region according to Equation (14). Red dot: shepherd; grey dots: sheep; blue dot: goal center. Blue dashed line: direction from steering agent to goal; red dashed lines: directions from steering agent to two sheep; angles shown as blue curves between the blue line and two red lines: minimum and maximum values. Time t is omitted here and in subsequent illustrations.

agents guiding multiple swarms through coordination between steering agents and strategies for recognizing swarms. The overall concept in constructing the algorithm is for each steering agent to sense swarms into one or multiple masses based on the angular difference to individuals and guide each mass sequentially using the orientation and driving stages. Orientation, where the agent moves itself behind the swarm relative to the goal, and driving, where it guides the swarm by switching between its border directions. The movement of steering agents relative to the mass is simplified by determining whether to move to the direction of the left or right boundary using reasonable rules. The limited angular information to individuals is sufficient to avoid disturbance from steering agents to individuals inside the swarms.

3.3.1 Single Steering Agent Guiding of One Swarm

In this part of the study, we use a reduction method that allows the steering agent to regard each swarm as a mass rather than a set of individuals. Our algorithm includes two stages, i.e. orienting behind the swarm relative to the goal, and driving the swarm towards the goal by switching between the two directions of the swarm borders. Compared to previous two-staged algorithms in shepherding [34, 27, 39], the proposed algorithm solely requires the steering agent to have bearing measurements, and the target swarm is modeled using a nonlinear agent-based model as denoted in Equation (2). Specifically, we judge these two stages by examining angle $\Theta_{q_k(t)}(c_k(t), x_g)$ and comparing it against a threshold θ_{orient} using the following expression:

$$|\Theta_{q_k(t)}(c_k(t), x_g)| < \theta_{\text{orient}} \tag{16}$$

where a result of 'false' indicates orientation, and a result of 'true' indicates driving. θ_{orient} is chosen to be sufficiently small enough to prevent judging two separate swarms into one, yet large enough to ensure that the steering agent does not mistakenly consider one swarm as multiple masses, particularly when the steering agent is close to the swarm. Notably, during the guidance process, the steering agent judges the situation defined in Equation (16) and decides between orientation and driving stage in each time step t. Figure 17 illustrates the circumstances of each of the two stages.

Let us first introduce notations commonly used in both stages. Depending on which side has the larger angle to the goal direction, we define unit vectors $\alpha_k(t)$ and $\alpha'_k(t)$ as

$$\alpha_k(t) = \begin{cases} \phi(p_{k_r}(t) - q_k(t)), & \text{if } \left|\Theta_{q_k(t)}(p_{k_r}(t), x_g)\right| \geq \left|\Theta_{q_k(t)}(p_{k_l}(t), x_g)\right|, \\ \phi(p_{k_l}(t) - q_k(t)), & \text{otherwise,} \end{cases} \tag{17}$$

and

$$\alpha'_k(t) = \begin{cases} \phi(p_{k_r}(t) - q_k(t)), & \text{if } \left|\Theta_{q_k(t)}(p_{k_r}(t), x_g)\right| < \left|\Theta_{q_k(t)}(p_{k_l}(t), x_g)\right|, \\ \phi(p_{k_l}(t) - q_k(t)), & \text{otherwise,} \end{cases} \tag{18}$$

where $\alpha_k(t)$ represents the direction where the angle relative to the goal direction is larger compared to the other direction, while $\alpha'_k(t)$ represents the other direction. We then let a unit vector $v_{k1}(t)$ represent the primary direction of movement for the shepherd. The vector $v_{k1}(t)$ is assigned a value equal to either $\alpha_k(t)$ or $\alpha'_k(t)$, which is determined by the following algorithm. Meanwhile, the steering agent receives repulsion from the direction of estimated swarm centre $c_k(t)$ and goal x_g. The velocity of shepherd k is then derived to be

$$v_k(t) = d_q\phi(d_1 R(\theta_1, v_{k1}(t)) + d_2\phi(q_k(t) - c_k(t)) + d_3\phi(q_k(t) - x_g)) \tag{19}$$

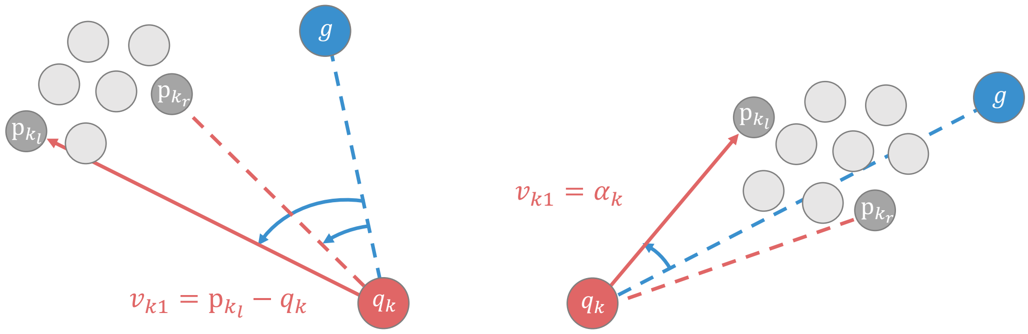
where d_1, d_2, and d_3 are positive constants, of which d_1 is larger than the others. Meanwhile, $R(\theta_1, v_{k1}(t))$ is an operator defined as

$$R(\theta_1, v_{k1}(t)) = \begin{cases} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} v_{k1}(t), & \text{if } v_{k1}(t) = \phi(p_{k_r}(t) - q_k(t)), \\ \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} v_{k1}(t), & \text{otherwise.} \end{cases} \tag{20}$$

to rotate vector $v_{k1}(t)$ by a non-negative angle θ_1. The direction of rotation depends on which side $v_{k1}(t)$ lies: if on the right side, then rotate right; and if on the left side, then rotate left. Equation (20) is used to rotate the movement of the steering agent away from the swarm to avoid collision risk.

Given that the steering agent cannot measure the distance to the other agents, moving directly toward the swarm may result in collisions. In cases wherein the angle $|\Theta_{q_k(t)}(c_k(t), x_g)|$ between the direction from shepherd k to the swarm and the direction from shepherd k to goal x_g is very large, as indicated by a result of 'false' in Equation (16), the steering agent needs to orient behind the swarm relative to the goal to reduce the angular difference $|\Theta_{q_k(t)}(c_k(t), x_g)|$.

Specifically, we choose the direction having the larger angular difference between the left and right sides of the swarm and denote it as $v_{k1}(t)$, i.e., we let $v_{k1}(t) = \alpha_k(t)$.



(a) Orientation stage (b) Driving stage

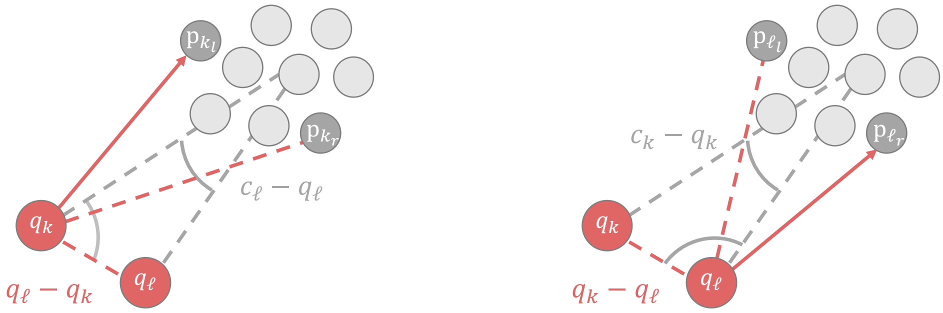
Figure 17. Illustration depicting a steering agent guiding a swarm to the goal region. The shepherding process is divided into two stages: orientation (Figure 17a) and driving (Figure 17b). Red solid line with an arrow: movement direction of a shepherd. During the orientation stage, a steering agent moves toward the sheep $p_{k_l}(t)$ or $p_{k_r}(t)$ according to the unit vector $\alpha_k(t)$ determined by comparing angles (indicated by blue) as in Equation (17) is larger than the other indicated by blue curves. During the driving stage, a steering agent continuously moves toward one passive agent until the angle indicated by a blue curve is small enough.

Throughout the orientation stage, as the steering agent moves to one side, the angle on that side increases, and thus the steering agent is expected to continuously move to the same side of the swarm based on the comparison of angles in Equation (17).

After skipping or completing the orientation stage, which is indicated by a result of 'true' in Equation (16), the steering agent begins to drive the swarm by alternately switching between the right and left sides. Based on $\alpha_k(t)$ defined in Equation (17), we define another unit vector as

$$\tilde{\alpha}_k(t) = \begin{cases} \phi(p_{k_r}(t) - q_k(t)), & \text{if } \alpha_k(t-1) = \phi(p_{k_r}(t-1) - q_k(t-1)), \\ \phi(p_{k_l}(t) - q_k(t)), & \text{otherwise,} \end{cases} \tag{21}$$

which remains at the same right or left side as that in the previous time step $t-1$ when $t > 0$ and keeps the same value as $\tilde{\alpha}_k(t) = \alpha_k(t)$ when $t = 0$. Then, similar to how we defined Equation (18), we denote a unit vector on the other side as $\tilde{\alpha}'_k(t)$.



(a) Shepherd k (b) Shepherd ℓ

Figure 18. Illustration depicting two steering agents cooperating to guide a swarm. Knowledge and next movements of steering agents k (Figure 18a) and l (Figure 18b) are shown. Grey dashed lines: directions from each steering agent to their respective estimated centers, which are shared between both steering agents; angles shown as grey curves: for calculation of their next movements, in accordance with Algorithm 1; red solid lines with arrows: movement of shepherd k to its left and shepherd ℓ to its right at next time step.

We then let $v_{k1}(t)$ be

$$v_{k1}(t) = \begin{cases} \tilde{\alpha}'_k(t), & \text{if } \operatorname{sgn} \angle(\tilde{\alpha}_k(t), \tilde{\alpha}'_k(t)) = \operatorname{sgn} \angle(\tilde{\alpha}_k(t), x_g - q_k(t)) \\ & \quad \text{and } |\angle(\tilde{\alpha}_k(t), x_g - q_k(t))| < \theta_{\text{drive}} \\ \tilde{\alpha}_k(t), & \text{otherwise} \end{cases} \tag{22}$$

where the operator sgn indicates whether a value is positive or negative, and θ_{drive} is a positive constant. At each time step, $v_{k1}(t)$ is adjusted to remain on the same side as that in the previous time step unless it is already at the edge of that side as determined by the angle conditions in Equation (22). Throughout the driving stage, the steering agent is expected to move to one of the right or left sides for a while, then switch to the other side and repeat this switching movement.

Algorithm 1 Multi-Shepherd coordination algorithm

Require: $M'_k(t), v_{k1}(t)$
Ensure: $M'_k(t) \neq \emptyset$
$C^l_k \leftarrow 0, C^r_k \leftarrow 0$ ▷ Initialise counters for left and right directions
while $\ell \in M'_k(t)$ **do** ▷ Increment counters if conditions are met
  if $|\angle(\phi(c_k(t) - q_k(t)), \phi(c_\ell(t) - q_\ell(t)))| < \theta_{n1}$ and $\left|\Theta_{q_k(t)}(p_{k_r}(t), p_{k_l}(t)) - 2\left|\Theta_{q_k(t)}(q_\ell(t), c_k(t))\right|\right| < 2\theta_{n2}$ **then**
    if $\Theta_{q_k(t)}(c_k(t), q_\ell(t)) < 0$ **then**
      $C^r_k \leftarrow C^r_k + 1$
    else
      $C^l_k \leftarrow C^l_k + 1$
    end if
  end if
end while
if $v_{k2}(t) = \phi(p_{k_r}(t) - q_k(t))$ and $C^l_k > C^r_k$ **then** ▷ Update by comparing counters
  $v_{k2}(t) \leftarrow \phi(p_{k_l}(t) - q_k(t))$
else if $v_{k2}(t) = \phi(p_{k_l}(t) - q_k(t)))$ and $C^r_k > C^l_k$ **then**
  $v_{k2}(t) \leftarrow \phi(p_{k_r}(t) - q_k(t))$
end if

3.3.2 Multiple Steering Agents Guiding of One Swarm

In the next part of this study, we increase the number of steering agents to perform the shepherding task more effectively. To avoid moving repeatedly among multiple steering agents, we design a strategy for coordination between two steering agents, denoted as k, ℓ, and subsequently apply it to more steering agents.

We allow communication between the steering agents because relying solely on the direction from shepherd k to shepherd ℓ is not sufficient for accessing their orientation relative to the swarm for coordination. Specifically, for shepherd k, we define the shared information as the direction from another shepherd ℓ to its estimated swarm centre $c_\ell(t)$. This additional knowledge is denoted as

$$Q_{k\ell}(t) = \{\phi(c_\ell(t) - q_\ell(t))\}. \tag{23}$$

Subsequently, we update the knowledge of shepherd k in Equation (15) to obtain the orientation of all the other steering agents as

$$\bar{Q}_k(t) = Q_k(t) \cup \bigcup_{\ell \in \mathcal{M}'_k(t)} Q_{k\ell}(t). \tag{24}$$

Based on the updated knowledge $\bar{Q}_k(t)$, we extend the proposed algorithm to allow for multiple steering agents guiding one swarm. Each steering agent independently decides its current stage. During the orientation stage, each steering agent determines its direction without considering the presence of other steering agents. During the driving stage, the movement of each steering agent is adjusted to avoid repeated movements with other steering agents. Specifically, we propose Algorithm 1 to modify $v_{k1}(t)$, where each steering agent can estimate the number of steering agents on its potential path to the right or left and choose to move to the side that has fewer steering agents. This estimation relies on comparing angles between specific directions observed by each steering agent and directions shared among them, as denoted in Equation (24). Specifically, for shepherd k and ℓ, the angle compared with θ_{n1} evaluates whether $q_k(t)$ and $q_\ell(t)$ have similar directions to the swarm center, while the angle compared with θ_{n2} evaluates whether $q_\ell(t)$ and the swarm center have similar directions to $q_k(t)$. If both conditions are met, shepherd ℓ is judged to be near the potential movement path of shepherd k. Shepherd k then determines whether that shepherd ℓ is on the left or right path. θ_{n1} and θ_{n2} are fixed values. Additionally, although the vectors $\phi(c_k(t) - q_k(t))$ and $\phi(c_\ell(t) - q_\ell(t))$ are pointing to different estimated swarm centers, we consider the error to be negligible. Figure 18 illustrates the shared information and movements of two steering agents during the driving stage.

3.3.3 Multiple Steering Agents Guiding of Multiple Swarms

In this part of the study, we place multiple swarms separately in their initial placements. Each steering agent lacks knowledge of the number of swarms based on its knowledge $\bar{Q}_k(t)$ defined in Equation (24). Instead, it observes these swarms as subswarms by comparing the angles of the interval between passive agents with a threshold θ_n. Therefore, the set of passive agents observed under occlusion by shepherd k, denoted as $\mathcal{N}'_k(t)$, is partitioned into multiple subswarms. Specifically, we partition set $\mathcal{N}'_k(t)$ to each subswarm τ as $\mathcal{N}'_k(t) = \bigcup_\tau \mathcal{N}'_{k\tau}(t)$ in such a way that any pair of passive agents $i \in \mathcal{N}'_{k\tau}(t)$ and $j \in \mathcal{N}'_{k\tau'}(t)$ satisfies $|\Theta_{q_k(t)}(p_i(t), p_j(t))| > \theta_n$ if and only if $\tau \neq \tau'$.

Coordination between the steering agents can be established by letting each steering agent sequentially target a specific subswarm rather than all subswarms. We first label the subswarm that has the largest absolute angle between the direction to its estimated center and the direction to the goal as

$$\mathcal{N}'^{\max}_k(t) = \max_\tau \left|\Theta_{q_k(t)}(c_{k\tau}(t), x_g)\right| \tag{25}$$

where $c_{k\tau}(t)$ represents the estimated centre of subswarm τ observed by shepherd k. We then let the steering agent execute the algorithm by observing passive agents in

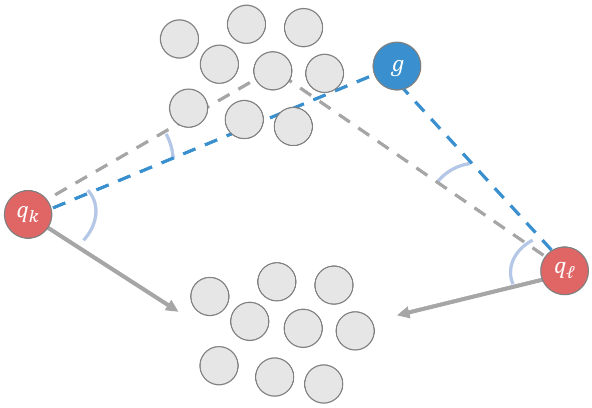


Figure 19. Illustration depicting two steering agents guiding two swarms by targeting a subswarm. Angles shown as blue curves: for comparison between each steering agent to choose a subswarm having a larger angle; grey solid lines with arrows: the subswarm targeted by both steering agents (herein, both are towards the bottom swarm) based on Equation (25).

the subset $\mathcal{N}_k^{\prime\max}(t)$ rather than $\mathcal{N}_k'(t)$. Following this strategy, each steering agent chooses and herds its target subswarm $\mathcal{N}_k^{\prime\max}(t)$ at each time step until all the swarms are inside the goal region. Figure 19 illustrates how two steering agents choose their target subswarms.

Meanwhile, each steering agent is not aware of whether other steering agents are targeting the same subswarm and needs only to estimate if the others are on its potential path, in accordance with Algorithm 1, to cooperate.

Summary. Through the descriptions presented earlier, we have proposed an algorithm by which varying numbers of steering agents can cooperatively guide multiple swarms in different distributions. The steering agents follow the final design of the algorithm regardless of their placements in the experiments.

3.4 Experiments

Our aim in experiments is to test whether shepherding can be successful and the essential amount of information for the proposed algorithm. For the experiment, we first assign values to the parameters of the passive agent model and shepherding algorithm, set up the initial placements of the experiments, and design metrics to evaluate the performance of the shepherding algorithm. Subsequently, we present simulation results that illustrate the trajectory and time-series variations for a single trial, and an evaluation across multiple trials. Finally, we reduce the angular accuracy of the bearing measurements and remove communication between the steering agents to conduct experiments to measure the information required to accomplish the shepherding task.

3.4.1 Parameter Values

In this part of the study, we select different parameter values and placements for the passive agents and observe their movements under these settings to determine an appropriate goal radius for shepherding. Additionally, we assign the parameter values for the shepherding algorithm and explain their rationales.

Because different parameter values for the same passive agent model can result in a variety of behaviors, we conduct experiments under several sets of parameter values c_1, c_2, c_3, c_4, c_5 and compare their levels of performance. Here, based on the movement characteristics of swarm systems [49], we present three sets of parameter values for the passive agents, as follows:

$$\begin{aligned} &c_1 = 200, c_2 = 0.2, c_3 = 0.2, c_4 = 1000, c_5 = 0.1, r = 60, \\ &c_1 = 250, c_2 = 0.15, c_3 = 0.2, c_4 = 1200, c_5 = 0.2, r = 60, \\ &c_1 = 200, c_2 = 0.2, c_3 = 0.25, c_4 = 800, c_5 = 0.05, r = 60, \end{aligned} \tag{26}$$

where we classify the first set as the default; and the second set as sensitive, owing to its larger separation c_1, smaller cohesion c_2, tendency to separate under larger repulsion c_4, and larger noise c_5; and the third set as insensitive, owing to its tendency to align with the others under larger alignment c_3, smaller repulsion c_4, and smaller noise c_5. The parameter value for alignment c_3 is overall increased to mimic swarm alignment behaviors. Additionally, we set $\delta = 3$ in Equation (4).

The size of the goal radius greatly influences the performance of the shepherding algorithm, particularly when The proposed algorithm relying on bearing measurements, is unable to accelerate shepherding with a small goal radius. This limitation arises from its inability to compress the area occupied by the swarm by moving closer to the agents due to its lack of proximity judgment. If the goal radius is much smaller than the radius of the swarm, the steering agents are likely to keep circling the goal point and its periphery without completing the task successfully. Alternatively, if the goal radius is set to be exceedingly large, the steering agents will easily complete the task, leaving us unable to assess the performance.

Therefore, we determine the goal radius by observing the shape of the swarm when it is relatively stationary without any steering agents or obstacles. Specifically, we measure the length of the swarm shape in terms of the maximum distance between agents, which is $x_s(t) = \max_{i,j \in \mathcal{N}(t)} \|p_i(t) - p_j(t)\|$. Then, by observing the time-series variation of $x_s(t)$, we regard that the swarm is stationary when $x_s(t)$ has little variation over time, i.e., if it satisfies $(1 - k_s)x_s(t_s + 1) < x_s(t_s) < (1 + k_s)x_s(t_s + 1)$ with $k_s = 0.02$ when $t \geq t_s$. We then define the goal radius $R_g = k_g x_s(t_s)$ with the coefficient $k_g = 0.8$. This procedure is followed to ensure a common goal radius for the subsequent experiments conducted under different sets of parameter values for the same placements, as outlined in Equation (26). When calculating the goal radius for multiple swarms, we consider that the steering agents must be able to collect all the swarms into the goal region. We first calculate the expected goal region for each swarm, then summarise the approximated goal radius for all passive agents in these swarms.

Additionally, given that steering agents rely solely on bearing measurements and cannot independently judge whether all the passive agents are inside the goal region, the completion of a shepherding task is determined externally and uniformly for all the steering agents.

For the parameter values of steering agents, we fix the magnitude of velocity for each steering agent at $d_q = 2$. This ensures that the velocity of the steering agents is moderately higher than that of the passive agents, as determined by the parameter values given in Equation (26). Additionally, regarding other coefficients appearing in Equation (19), we set d_1 to be much larger than d_2, d_3, as $d_1 = 5$, $d_2 = 1$, and $d_3 = 1$ to make $v_{k1}(t)$ the primary movement. We assign the sensing range $r' = 300$ to ensure that the range is sufficiently large for sensing other agents. We then assign the angle thresholds for each steering agent as $\theta_{\rm occ} = \pi/60$, $\theta_1 = \pi/9$, $\theta_{\rm orient} = \pi/3$, $\theta_{\rm drive} = \pi/18$, $\theta_{\rm n1} = \pi/4$, $\theta_{\rm n2} = \pi/2$, $\theta_n = \pi/6$. These angles are assigned appropriate values based on the following rationales: angle $\theta_{\rm occ}$ is set to a small value to imitating observation under occlusion; angle θ_1 is moderately adjusted to avoid collisions between the steering agents and swarms; angle $\theta_{\rm orient}$ is limited to no more than $\pi/2$ to determine the orientation or driving stage; angle $\theta_{\rm drive}$ is appropriately small to allow complete driving on one side before switching to the other side; angles $\theta_{\rm n1}, \theta_{\rm n2}$ are chosen reasonably to determine whether there are other steering agents on their paths; and angle θ_n is appropriately small to ensure correct recognition of subswarms.

For this next part of the experiment, we design three initial placements of the passive agents and steering agents where N passive agents are distributed into n different swarms. Specifically, each swarm, denoted by σ, consists of N_σ passive agents that are randomly placed on a disk centered at each origin with an initial radius of $R_{s\sigma}$ when $t = 0$. We denote the numbers and radii of multiple swarms as follows:

- $N_1 = 30, R_{\rm s1} = 40$,
- $N_1 = 30, R_{\rm s1} = 40$ and $N_2 = 50, R_{\rm s2} = 60$,

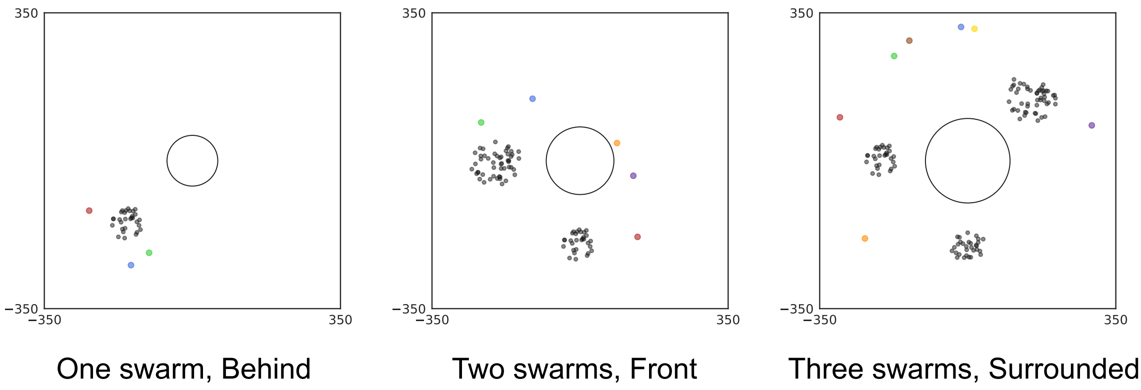


Figure 20. Initial placements of swarms, steering agents, and goal. Positions of passive agents vary from one to three swarms around the goal. Numbers of steering agents are $M = 3$, $M = 5$, and $M = 7$, respectively. Positions of steering agents are categorized as being behind the swarms, in front of the swarms, or surrounding the swarms. Shepherds are behind the swarms if the steering agents are farther from the goal than the swarms, with similar directions to the goal; steering agents are in front of the swarms if the steering agents are close to the goal center; steering agents are around the swarms if the steering agents are farther from the goal than the swarms, with directions to the goal coming from all around.

- $N_1 = 30, R_{s1} = 40$, $N_2 = 30, R_{s2} = 40$ and $N_3 = 50, R_{s3} = 60$.

We then set up the placements between these swarms and between the swarms and the goal, and position the steering agents at various positions to the swarms relative to the goal, such as behind the swarms, in front of the swarms (near the goal), and surround the swarms, as illustrated in Figure 20.

We design the following two metrics to evaluate the effectiveness and stability of the proposed algorithm in the shepherding task. One metric measures progression in individual trials, whereas the other measures performance across multiple trials.

Time-series variations in distances over time: This metric calculates the distances from the passive agents and steering agents to the goal during each trial. In successful trials, we observe that the distance from each passive agent i to the goal, $|p_i(t) - x_g|$, decreases from an initial value to a value below the goal radius R_g. We record the mean value and the upper and lower intervals for the passive agents. Similarly, we denote the distance from shepherd k to the goal as $|q_k(t) - x_g|$. The mean value for the steering agents usually follows the values for the passive agents because the steering agents guide the passive agents to the goal region.

Consumed time: This metric measures the overall performance of the steering agents across multiple trials. The total time consumed in all trials is counted and used to draw box plots that visualize the experiment results. Low consumed time and small variation between trials indicate effective and robust shepherding.

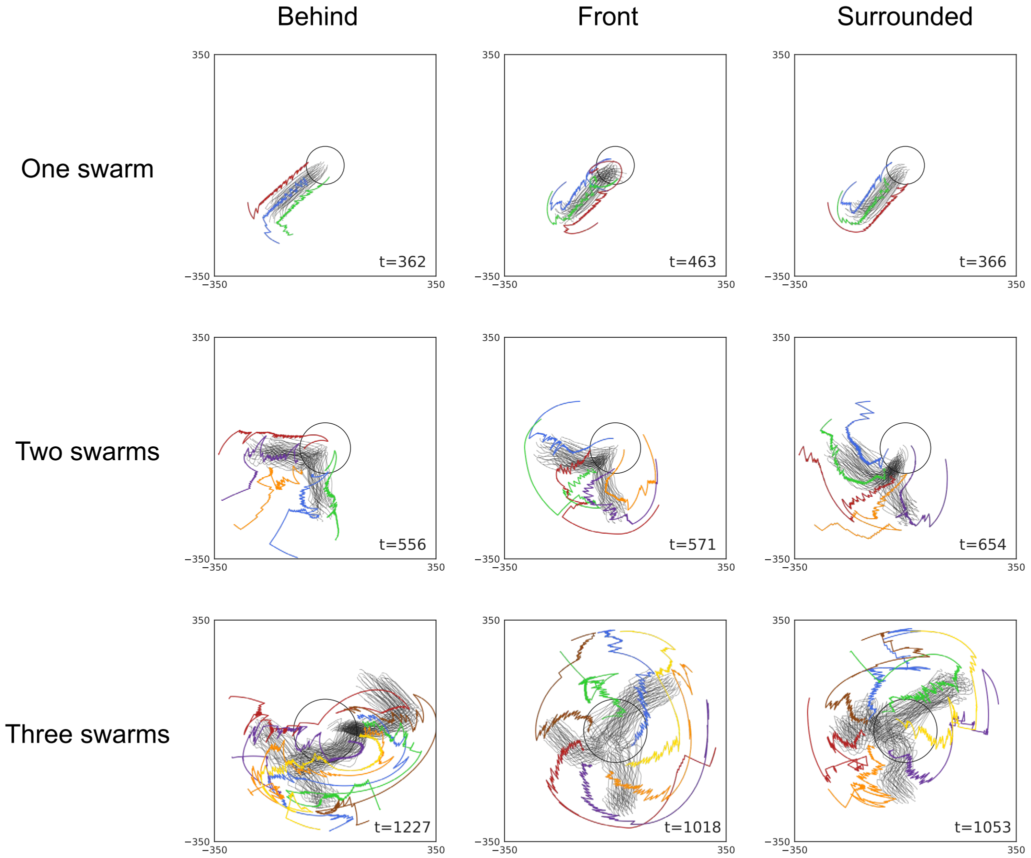
3.4.2 Experiment Results

We conduct experiments in $C = 20$ trials with an upper time limit T for each placement. The shepherding succeeds only when the consumed time is shorter than the upper limit T. Specifically, we set $T = 3000$ to ensure sufficient time steps to complete the shepherding task. We conduct simulation experiments for three placements using the appropriate number of steering agents and the default parameter values for passive agents. For the tasks of shepherding a swarm, two swarms, and three swarms, we show the trajectories in Figure 21a and numerically illustrate the shepherding process in Figure 21b by displaying the time-series variation in the distances of the passive agents and steering agents to the goal for a random trial. We then specifically show the simulation results in increasing the number of steering agents for shepherding two swarms in Figure 22 and three swarms in Figure 23.

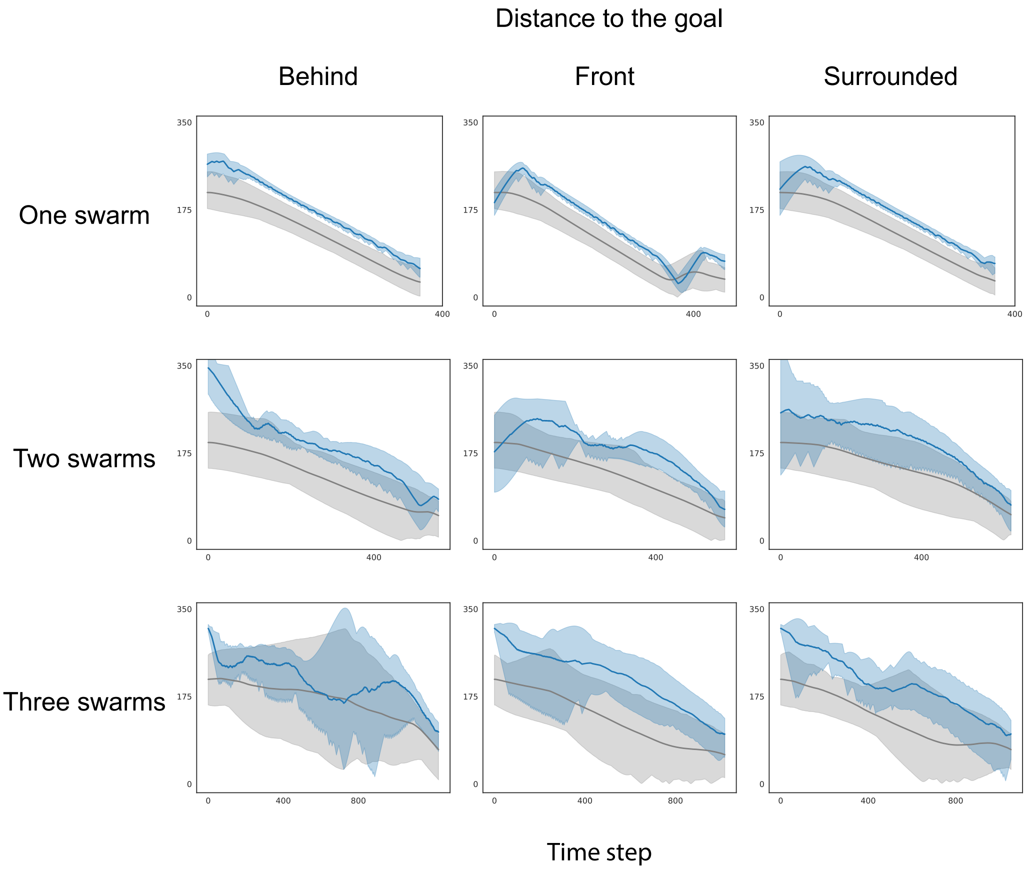
From the trajectories, we observe that the movement of each steering agent is practically divided into two stages and repeated several times, especially in the cases of shepherding three swarms, which aligns with the proposed algorithm that includes the orientation and driving stages. When multiple steering agents guide the same swarm, the movements at each time step indicate that the steering agents can recognize neighboring steering agents and avoid converging toward each other. This phenomenon naturally results in the steering agents dynamically encircling the target swarm and collectively driving it to the goal region. Furthermore, when guiding multiple swarms, each steering agent can estimate the orientation of other steering agents relative to its target subswarm and sequentially drive the swarms without duplicating movements with the other steering agents.

We then evaluate the shepherding performance given the initial placements described earlier and using three sets of parameter values for the sheep, with the number of steering agents increasing from 1 to 10. The results are presented as changes in consumed time across C trials. In Figure 24, to avoid redundancy, we present only the results for guiding two and three swarms. We observe that whereas the steering agents struggle to succeed with shepherding when their numbers are low, success rates increase and consumed time decreases as the number of steering agents increases, eventually reaching a 100% success rate and gradually decreasing the consumed time, which demonstrates the significance of communication. We then compare the differences in shepherding results among the three sets of parameter values. We note that guiding sensitive passive agents tends to fail, whereas guiding insensitive passive agents tends to succeed and consume less time. Nevertheless, the shepherding results generally remain stable regardless of changes in the parameter values for the passive agents.

Based on the results above, we have observed that the advantage of the proposed algorithm is that, although no communication is used, better performance is achieved by assigning the movement of each steering agent to different target passive agents. Specifically, the algorithm leverages cooperation among multiple agents using observable position information, even though the agents do not communicate with each other to share additional information. This approach improves scalability to changes in initial placements, accommodates increases in shepherd numbers, and enhances robustness to reductions in sensing accuracy.

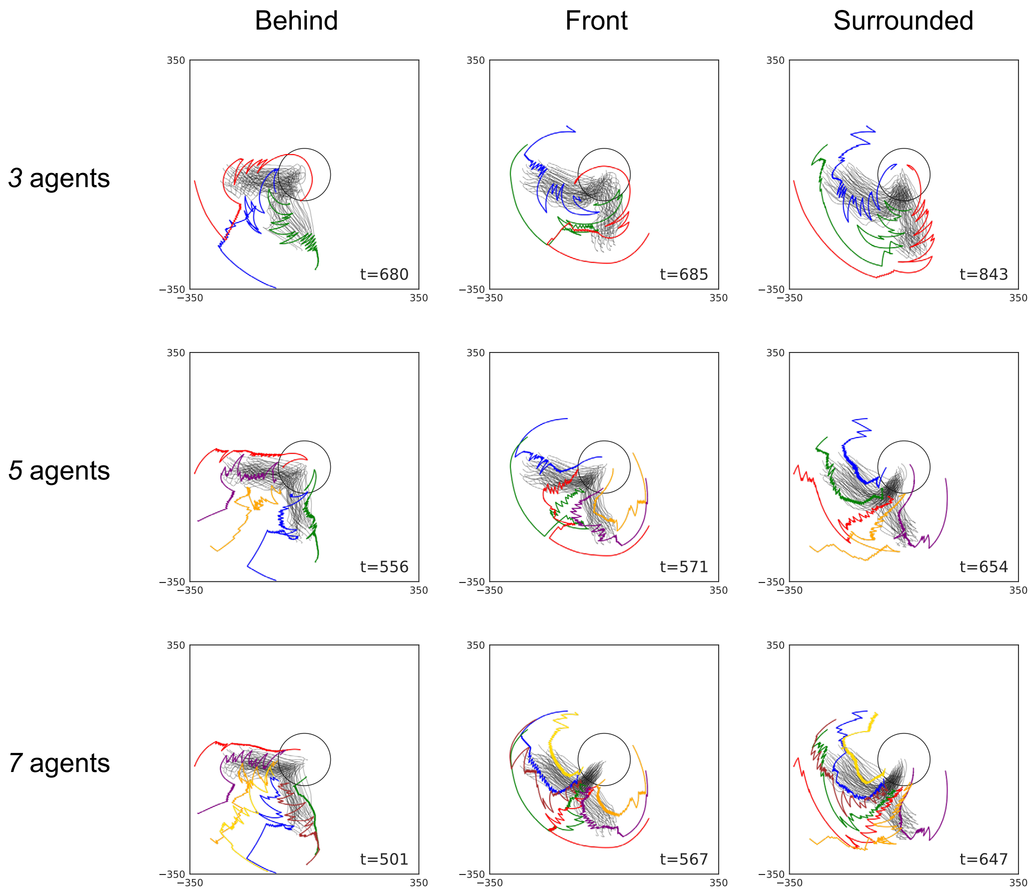


(a) Trajectories for guiding one to three swarms under three initial placements for steering agent. Trajectories are the same as the initial placements as in Figure 20.

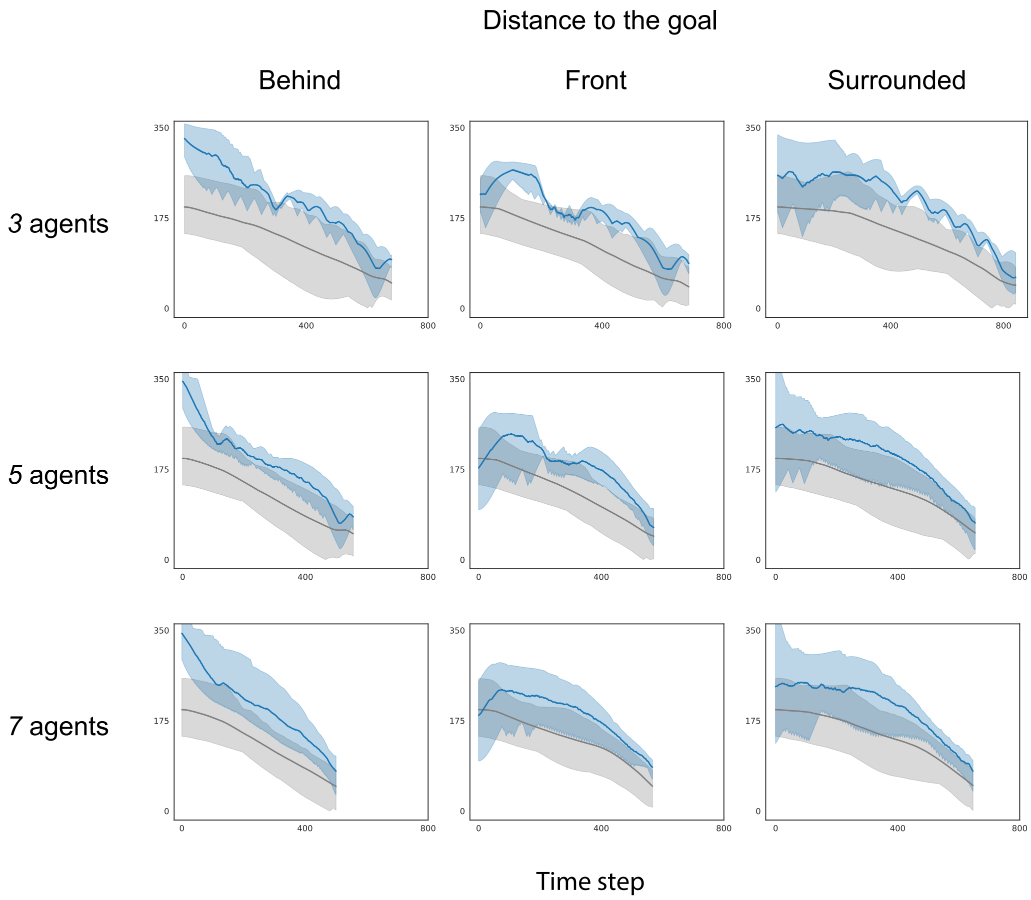


(b) Time-series variations in distance to the goal for passive agents and steering agents. Blue line and interval: steering agents; grey line and interval: passive agents.

Figure 21. Trajectories and time-series variations for guiding one to three swarms under three types of initial placements for steering agents, using the default set of sheep-model parameter values. Numbers of steering agents are $M = 3$, $M = 5$, and $M = 7$ for the case of one, two, and three swarms, respectively.

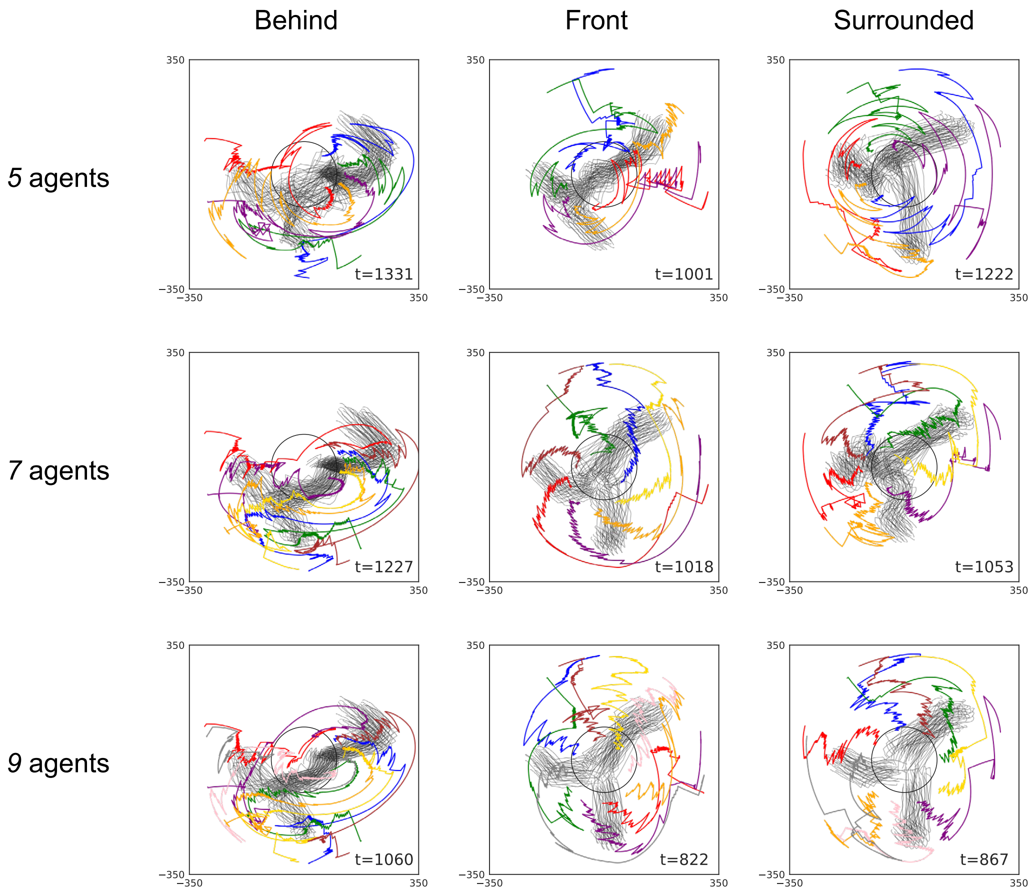


(a) Trajectories for guiding two swarms with an increasing number of steering agents.

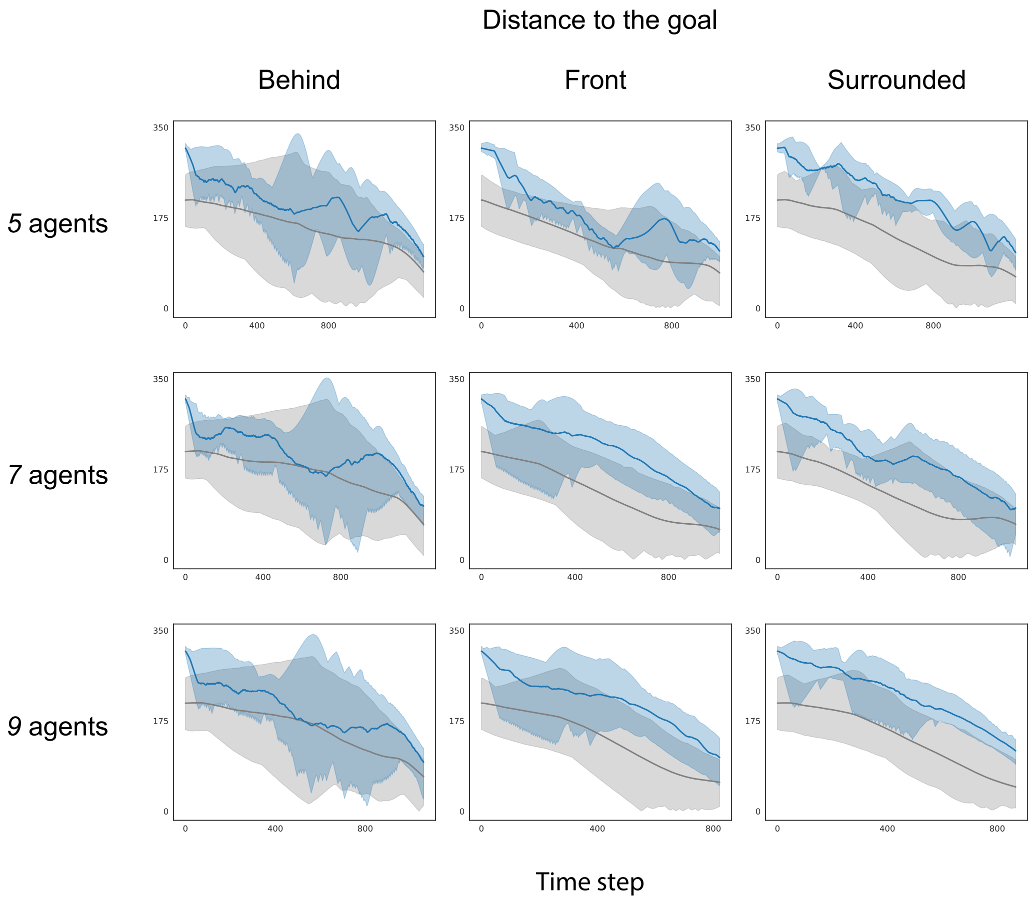


(b) Time-series variations in distance while guiding two swarms with an increasing number of steering agents. Blue line and interval: steering agents; grey line and interval: passive agents.

Figure 22. Trajectories and time-series variations for guiding two swarms with an increasing number of steering agents under three types of initial placements for steering agents, using the default set of sheep-model parameter values. Numbers of steering agents are $M = 3$, $M = 5$, and $M = 7$, respectively.

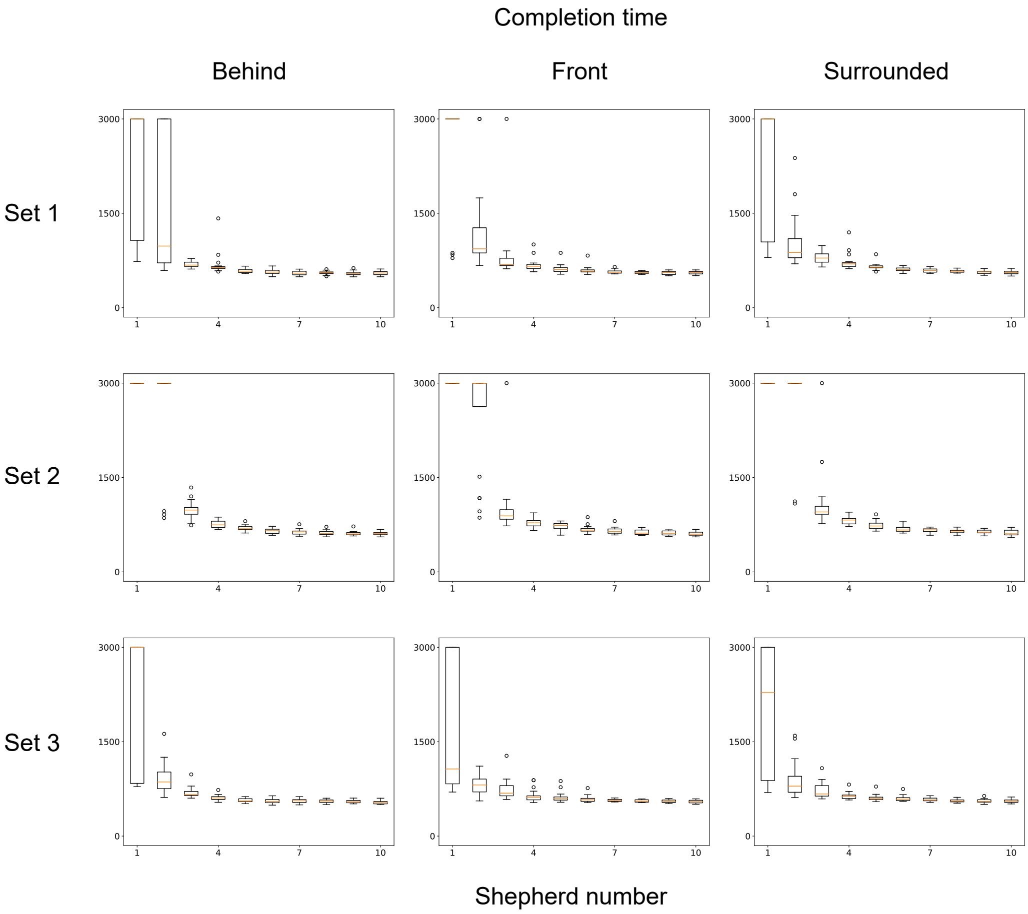


(a) Trajectories for guiding three swarms with an increasing number of steering agents.

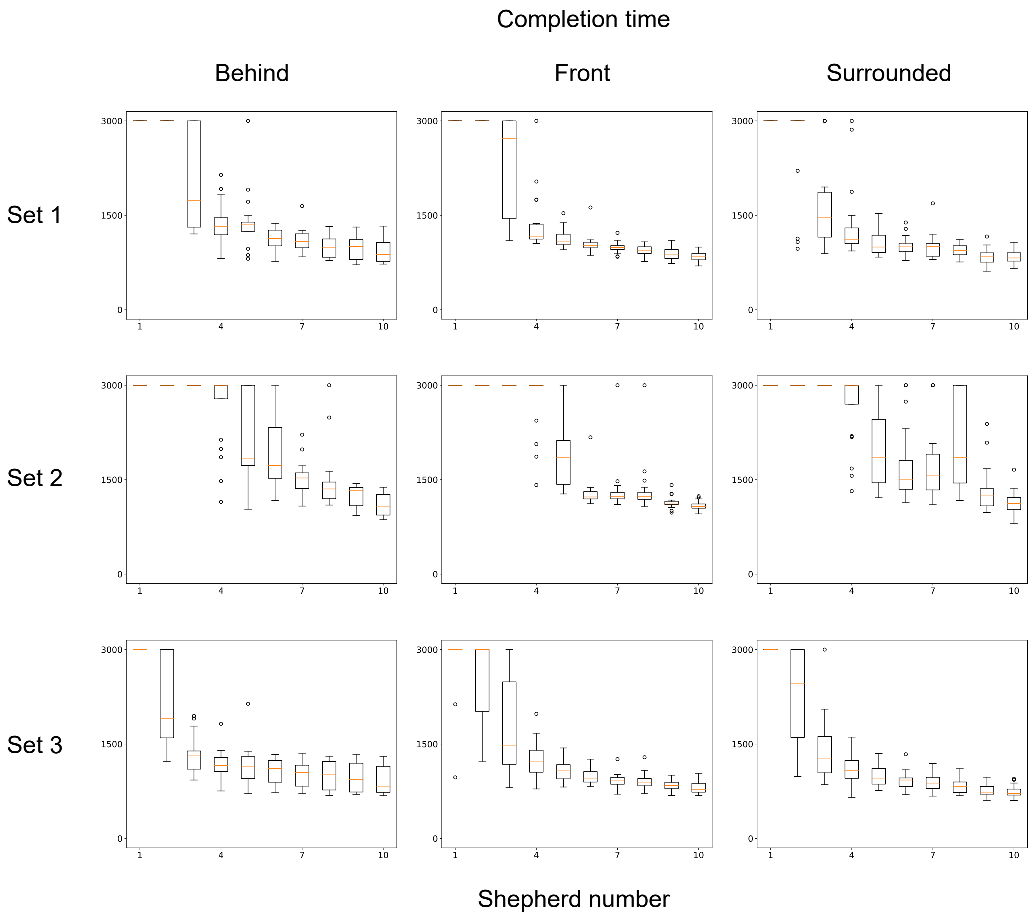


(b) Time-series variations in distance while guiding three swarms with an increasing number of steering agents. Blue line and interval: steering agents; grey line and interval: passive agents.

Figure 23. Trajectories and time-series variations for guiding three swarms with an increasing number of steering agents under three types of initial placements for steering agents, using the default set of sheep-model parameter values. Numbers of steering agents are $M = 5$, $M = 7$, and $M = 9$, respectively.



(a) Box plots of time consumed guiding two swarms with respect to the number of steering agents.



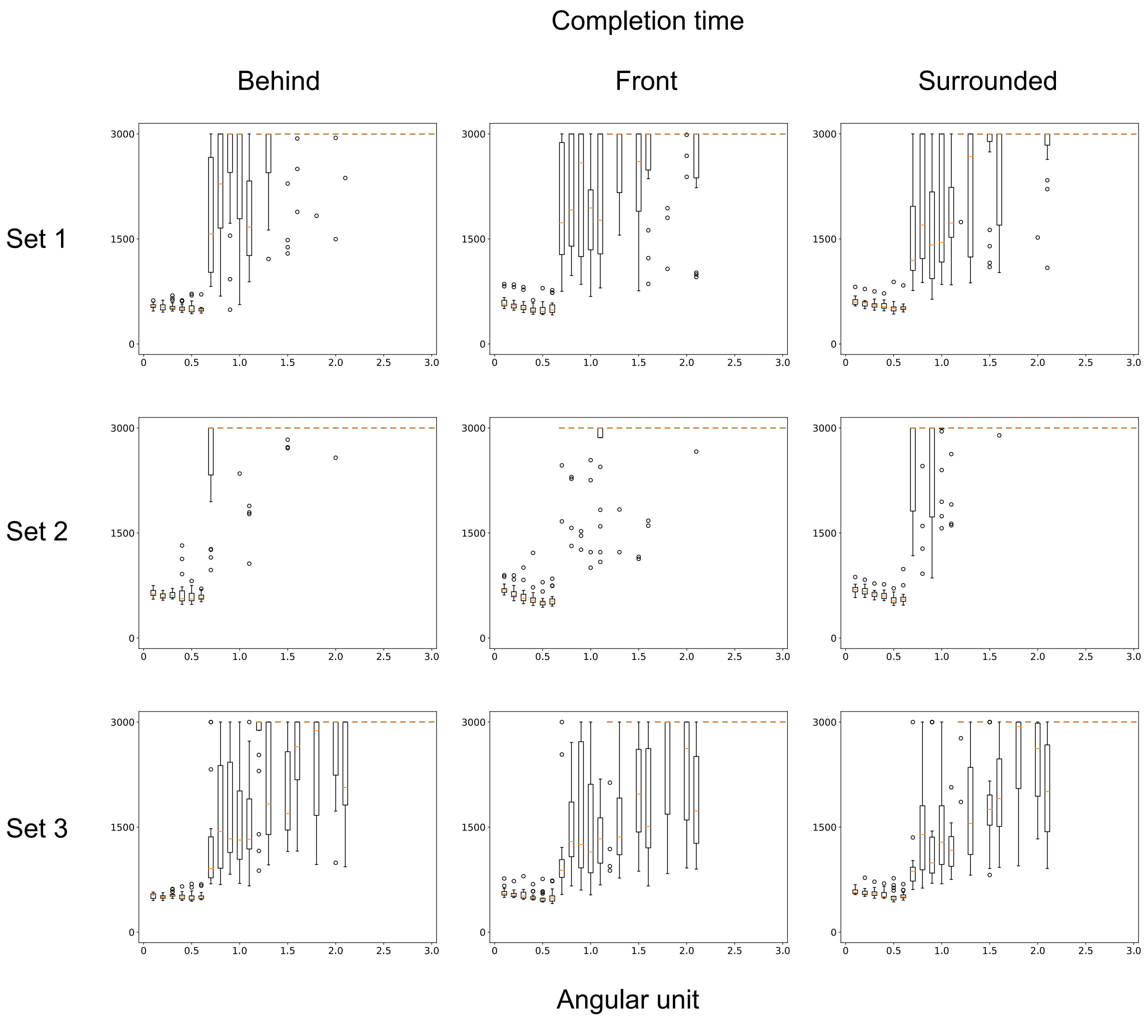
(b) Box plots of time consumed guiding three swarms with respect to the number of steering agents.

Figure 24. Box plots of time consumed guiding two and three swarms with respect to the number of steering agents, for three sets of sheep-model parameter values. The number of steering agents M varies from 1 to 10, for three sets of sheep-model parameter values.

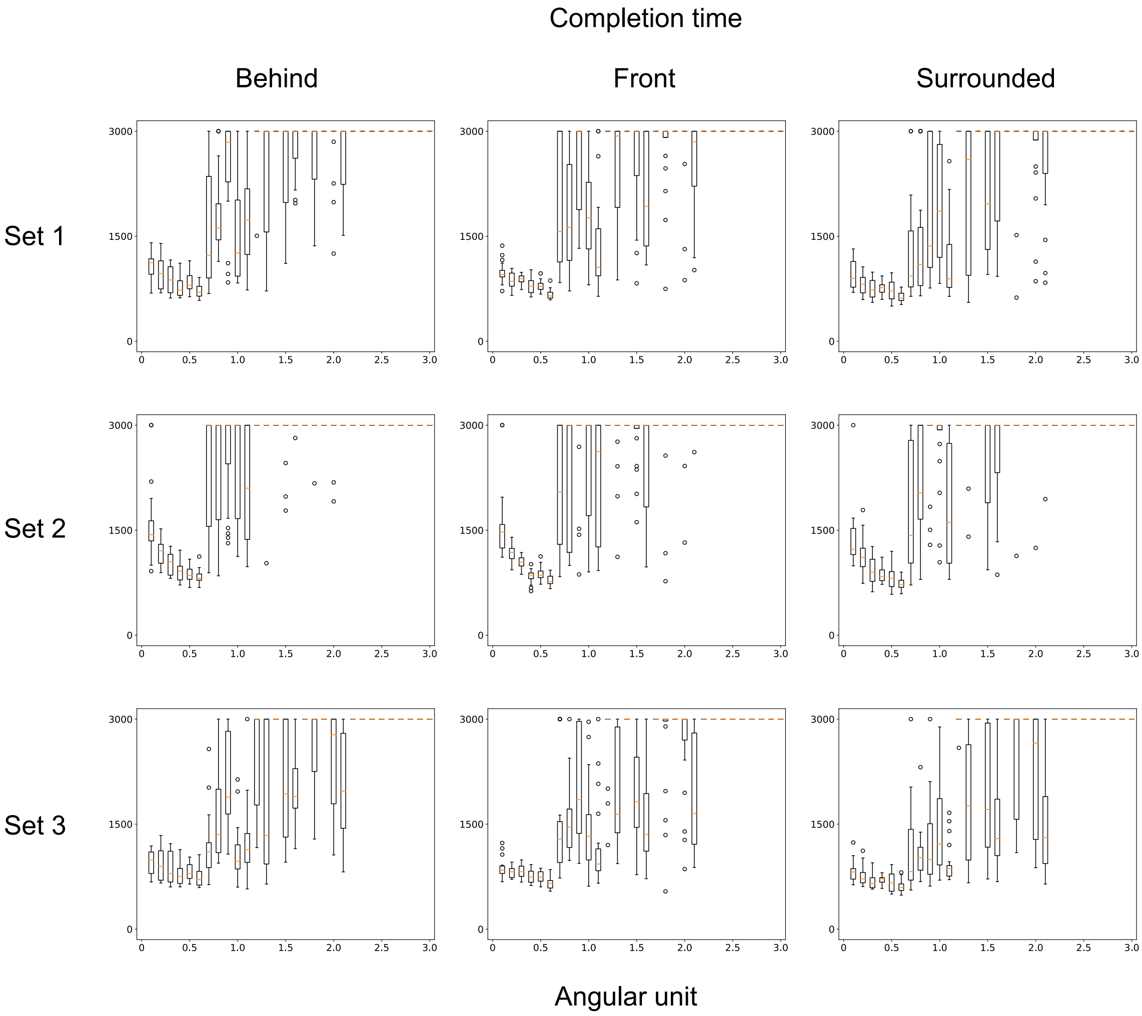
3.4.3 Experiments with Further Reduced Information

Furthermore, we investigate the essential amount of information, in terms of angular accuracy and communication, required for guiding. We determine that moderate angular accuracy and limited communication between steering agents are required for our bearing-only algorithm.

With regard to angular accuracy, we increase the error for any angle measured by rounding the value of each angle down to the nearest multiple of a unit, starting from a default with no error, ranging from a radian of 0.1 to 3, with increments of 0.1, as used in bearing measurements outlined in Equation (15). In the experiments, we assign $M = 7$ to be the number of steering agents that are to guide three swarms, which is sufficient for success with no angular error. The results are shown in Figure 25. We observe that when the error is small, the consumed time does not change much and may even become shorter until approximately 0.5. We believe that this phenomenon occurs because, under the assumption that there is no error in measuring the angles, the movements of the steering agents result in unnecessary reactions to minor changes in the angle. This oscillation decreases as the error increases. However, as the error continuously increases, the consumed time begins to fluctuate, and the success rate significantly decreases. Trajectories with increasing angular error are illustrated in Figure 26. The trajectory varies depending on whether the shepherding succeeds or fails. In successful trials, steering agents usually guide the swarms while maintaining the shape of the swarm until all the passive agents reach the goal region. On the other hand, in failed trials, the steering agents gradually lose precise control of the swarms as the angular error increases. With larger errors, the shape of the swarm may exceed the size of the goal region even if the steering agents continue circling the swarm to guide it into the goal region. With even larger errors, the swarm may become completely fragmented and scattered by the steering agents.



(a) Box plots of time consumed guiding two swarms with respect to angular error. Number of steering agents is set to $M = 5$.



(b) Box plots of time consumed guiding three swarms with respect to angular error. Number of steering agents is set to $M = 7$.

Figure 25. Box plots of time consumed guiding two and three swarms with respect to angular error, for three sets of sheep-model parameter values. The angular error increases from 0 by 0.1 to 3.

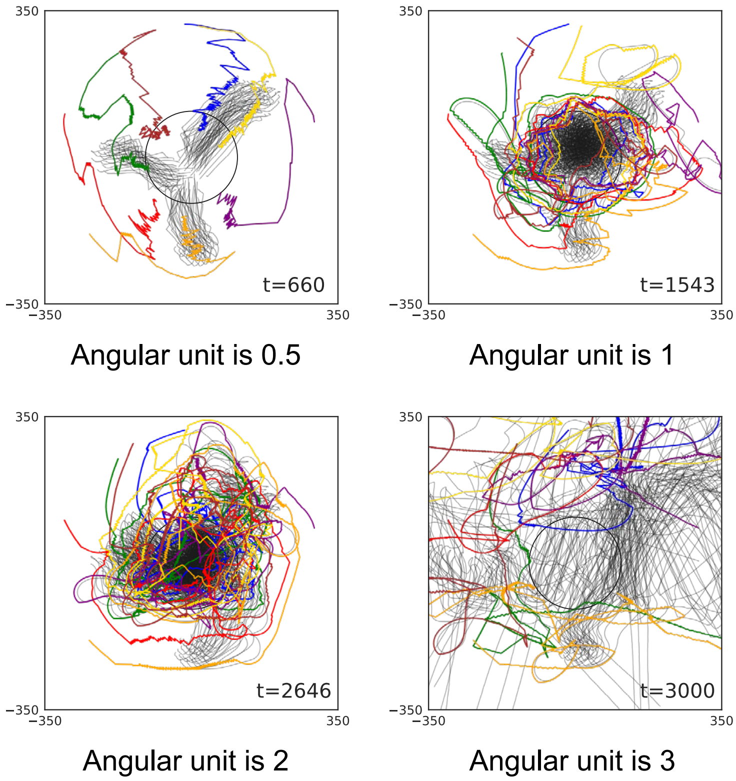


Figure 26. Trajectories of guiding three swarms as angular unit increases from 0.5, to 1, 2, and 3, respectively, for default set of sheep-model parameter values. Initial placements are the same as in Figure 20.

With regard to communication, the proposed algorithm requires only the information $Q_{k\ell}(t)$ given in Equation (23). If we attempt to remove the only communication between each pair of steering agents, there would be no coordination among the steering agents, potentially leading to collisions and overlapping movements. An example of shepherding without communication is illustrated in Figure 27 where steering agent number $M = 3$ for guiding one swarm, $M = 5$ for guiding two swarms, and $M = 7$ for guiding three swarms. The trajectories of steering agents become repetitive due to the lack of communication, which prevents accounting for the presence of other steering agents, thus failing to differentiate their movements. As the effectiveness does not improve with an increasing number of steering agents, this approach takes longer to complete and is more likely to fail when multiple swarms exist.

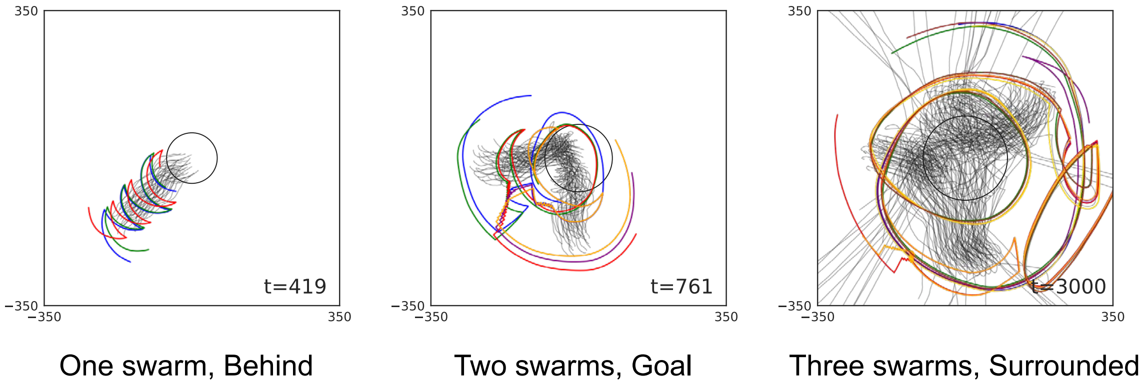


Figure 27. Trajectories of guiding one to three swarms with no communication between steering agents, for the default set of sheep-model parameter values. Initial placements, as well as the number of passive agents and steering agents, are the same as in Figure 20, and the method is compared with that for Figure 21a.

3.5 Summary

In this chapter, from the aspect of further reduced sensing information cases, we propose a shepherding algorithm that relies on bearing measurements and low-level inter-agent communication to achieve cooperation for successful shepherding. The approach emphasizes the collective movements of entire swarms rather than individual passive agents. Each steering agent is enabled to sense the orientation of the two boundaries of each swarm and recognize multiple swarms based on angular differences. Subsequently, we propose a generalized shepherding algorithm that does not require knowledge of the exact swarm model or individual passive agents. Additionally, we devise methods for steering agents to select target swarms and cooperate in the driving stage by confirming their relative orientations to the swarms. Experiments are conducted to evaluate performance under different placements and parameter values, demonstrating the effectiveness of the proposed algorithm with varying numbers of steering agents. Furthermore, we investigate the roles of angular accuracy and communication among steering agents in shepherding and the minimum conditions for both types of information required in shepherding.

The design of this shepherding algorithm based on bearing measurements draws inspiration from conventional shepherding algorithms, which divide the shepherding problem into manageable sub-problems. Although the information used decreases, the core strategy remains consistent and effective. However, the simulation has not been tested for scenarios where the number of sheep is significantly increased or parameter values are adjusted to make passive agents more prone to separating from the swarm. These cases reveal a bottleneck in performance.

Although the stability of shepherding using the bearing-based algorithm cannot be formally established due to the nonlinear and highly complex dynamics of swarms of passive agents, which make direct proof infeasible, a potential approach to address this challenge involves abstracting the swarm model into a single-agent model. Future research could focus on establishing stability using Lyapunov functions, constructing a mathematical model, and analyzing error states to demonstrate the stability of a system where a steering agent guides a single passive agent to a target point. Although the proposed shepherding algorithm relies on bearing measurements, the stability proof needs to account for relative positions and velocities. Such a proof would enhance the applicability and robustness of the proposed algorithm in practical scenarios.

Chapter 4

Conclusion

Throughout the two primary studies presented in this dissertation, we examine the key types of sensing and communication information required for cooperation among multiple steering agents to achieve swarm shepherding, which addresses the broader question of the key information requirements for swarm navigation. Through algorithmic development and extensive experiments, this research indicates how specific sensing and communication capabilities enable effective swarm control. Specifically, the findings from the first study revealed that the success of shepherding tasks depends not only on the amount of information but also on its nature, with factors such as relative positioning, rather than absolute coordinates, playing a pivotal role. Meanwhile, the second study demonstrates that effective guidance can still be achieved under constrained conditions, relying on bearing measurements and basic information sharing. These insights highlight the practical potential for reducing centralized information demands without significant performance losses, particularly in scenarios with limited bandwidth or sensor capabilities.

One of the crucial contributions of the dissertation is laying a framework for developing quantitative benchmarks under systematically varied sensing ranges, accuracies, and communication capabilities. These benchmarks will serve as practical references for designing swarm systems under real-world constraints, assisting engineers in balancing system performance with hardware limitations. Furthermore, the scalability of the algorithms under different information constraints will be demonstrated across a range of swarm sizes and placements. This adaptability supports their application in dynamic environments, where agents must effectively respond to changes in swarm distribution, environmental obstacles, and task requirements.

Extensive experiments in the first and second studies reveal that swarm guidance can still be achieved despite information constraints. For example, in cases of low sensing accuracy, the system remains functional although there may be trade-offs in time efficiency. This finding emphasizes the robustness of our proposed algorithms and suggests the resilience potential of systems with constrained sensing inputs. In situations where communication is limited or absent, our research demonstrates that swarm control can still be maintained, allowing each agent to assess the orientation of the swarm relative to its target independently. This approach is particularly relevant for applications in remote or hazardous locations where communication infrastructure may be unavailable, as well as in resource-limited robotic applications.

Although both of our proposed algorithms succeeded in their respective studies, the first communication-free algorithm demonstrates greater effectiveness and robustness due to its mechanism of targeting specific agents to drive the entire swarm. In contrast, the second bearing-based algorithm considers the swarm as a whole for decision-making and control. While both studies use the same Boid model for passive agents, the parameter values in the first study are more aligned with individual behaviors, whereas those in the second study are closer to collective dynamics, better reflecting the distinct characteristics of each algorithm. Furthermore, performance could be improved by combining the targeting strategy from the first study with the dividing-stage mechanisms from the second study, resulting in a more effective approach.

The methods proposed in this dissertation enhance our understanding of essential information for effective swarm guidance, while several promising research directions remain. Future studies could explore adaptive mechanisms, such as reinforcement learning, to enable real-time parameter tuning and decision-making. Equipping steering agents with the ability to autonomously learn and adapt to environmental changes could increase robustness across diverse conditions, reducing reliance on pre-configured settings. Additionally, hybrid architectures combining rule-based and learning-based control could optimize swarm behavior. For example, a hybrid model might follow predefined rules in routine scenarios but employ reinforcement learning in unpredictable situations to broaden the applicability of swarm control to complex, real-world tasks.

While our studies show that swarm control can operate effectively with limited sensing and communication, further exploration of communication protocols could facilitate the collection of additional critical information to enhance performance. Designing protocols prioritizing essential information, such as urgent positional shifts or environmental hazards, could enhance coordination in resource-constrained settings. This dissertation establishes a theoretical foundation for understanding how essential information governs swarm behavior, contributing valuable insights to the field of swarm dynamics. By reducing dependency on centralized control and extensive communication, we outline practical implications for the scalability and feasibility of swarm systems in various contexts. Grounded in principles of nonlinear control and collective behavior, this approach bridges theory and practical application, offering a pathway for future systems that are not only efficient but also robust and adaptable.

From the perspective of swarm movement models, this dissertation investigates the performance of shepherding algorithms on swarms of passive agents in the Boid model. We characterize the Boid-like properties by collective movements within the swarms and repulsive reactions to steering agents, both of which we believe are essential for shepherding. Similar properties are present in the Couzin model [50] and the Vicsek model [51]. These characteristics are fundamental to shepherding by allowing steering agents to guide swarms through localized stimuli and interactions. In other models exhibiting Boid-like properties, adjustments to the implementation might be required to accommodate different swarm dynamics. Nevertheless, the core concepts of our proposed algorithms remain valuable: targeting a single agent to drive the swarm and dividing the shepherding process into subproblems to enhance performance in complex scenarios. In contrast, shepherding fails when attempting to guide swarms that do not exhibit Boid-like properties, such as those following the random-walk model [52], even though each passive agent has repulsive reactions to the steering agents.

The result of this dissertation demonstrates the potential applicability of swarm shepherding algorithms across diverse domains. Future work could focus on deploying these algorithms in real-world scenarios for empirical validation and refinement. Cooperations with related fields may offer new perspectives and technical tools to enhance practical applications. As the demand for multi-objective tasks in robotics grows—such as simultaneous obstacle avoidance and goal-reaching—incorporating multi-objective optimization techniques could enable swarm systems to dynamically balance competing objectives, thereby expanding their range of applications.

The advancements presented in this dissertation mark a critical step toward intelligent swarm control in real-world applications. As swarm systems gain wider adoption, they are expected to revolutionize fields such as automated agriculture, forestry, disaster response, environmental conservation, and large-scale coordination of robots and vehicles. By addressing resource constraints, this research lays the groundwork for making swarm systems more accessible and affordable. Ultimately, this work advances swarm intelligence, positioning swarm systems as powerful tools to complement and extend human capabilities. With continued technological progress, swarm control is poised to redefine possibilities in robotics, artificial intelligence, and autonomous systems, transforming industries and enhancing the quality of life.

Bibliography

[1] C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, and G. Volpe, "Active particles in complex and crowded environments," *Reviews of Modern Physics*, vol. 88, no. 4, p. 045006, 2016.

[2] A. Cavagna and I. Giardina, "Bird flocks as condensed matter," *Annual Review of Condensed Matter Physics*, vol. 5, no. 1, pp. 183–207, 2014.

[3] C. M. Topaz, A. J. Bernoff, S. Logan, and W. Toolson, "A model for rolling swarms of locusts," *The European Physical Journal Special Topics*, vol. 157, pp. 93–109, 2008.

[4] F. Lu, M. Zhong, S. Tang, and M. Maggioni, "Nonparametric inference of interaction laws in systems of agents from trajectory data," *Proceedings of the National Academy of Sciences*, vol. 116, no. 29, pp. 14424–14433, 2019.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34, 1987.

[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] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.

[8] G. Sun, R. Zhou, Z. Ma, Y. Li, R. Groß, Z. Chen, and S. Zhao, "Mean-shift exploration in shape assembly of robot swarms," *Nature Communications*, vol. 14, no. 1, p. 3476, 2023.

[9] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[10] H. Xie, M. Sun, X. Fan, Z. Lin, W. Chen, L. Wang, L. Dong, and Q. He, "Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation," *Science Robotics*, vol. 4, no. 28, p. eaav8006, 2019.

[11] Q. Wang and L. Zhang, "External power-driven microrobotic swarm: from fundamental understanding to imaging-guided delivery," *ACS Nano*, vol. 15, no. 1, pp. 149–174, 2021.

[12] L. Gómez-Nava, R. Bon, and F. Peruani, "Intermittent collective motion in sheep results from alternating the role of leader and follower," *Nature Physics*, vol. 18, no. 12, pp. 1494–1501, 2022.

[13] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[14] S. Qu, M. Abouheaf, W. Gueaieb, and D. Spinello, "An adaptive fuzzy reinforcement learning cooperative approach for the autonomous control of flock systems," in *2021 IEEE International Conference on Robotics and Automation*, pp. 8927–8933, 2021.

[15] R. Goel, J. Lewis, M. A. Goodrich, and P. Sujit, "Leader and predator based swarm steering for multiple tasks," in *2019 IEEE international conference on systems, man and cybernetics*, pp. 3791–3798, 2019.

[16] J.-M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *IEEE International Conference on Robotics and Automation, 2004*, vol. 4, pp. 4159–4164, 2004.

[17] 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 Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 523–537, 2020.

[18] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[19] D. J. Sumpter, "The principles of collective animal behaviour," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 361, no. 1465, pp. 5–22, 2006.

[20] 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.

[21] K. J. Yaxley, K. F. Joiner, and H. Abbass, "Drone approach parameters leading to lower stress sheep flocking and movement: Sky shepherding," *Scientific Reports*, vol. 11, no. 1, p. 7803, 2021.

[22] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS One*, vol. 5, no. 4, p. e10047, 2010.

[23] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.

[24] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, p. eaat3536, 2018.

[25] F. Mou, X. Li, Q. Xie, J. Zhang, K. Xiong, L. Xu, and J. Guan, "Active micromotor systems built from passive particles with biomimetic predator–prey interactions," *ACS Nano*, vol. 14, no. 1, pp. 406–414, 2019.

[26] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *Journal of Vibration and Control*, vol. 18, no. 7, pp. 1056–1066, 2012.

[27] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, 2017.

[28] R. A. Licitra, Z. I. Bell, and W. E. Dixon, "Single-agent indirect herding of multiple targets with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 847–860, 2019.

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

[30] J. Hu, A. E. Turgut, T. Krajník, B. Lennox, and F. Arvin, "Occlusion-based coordination protocol design for autonomous robotic shepherding tasks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 1, pp. 126–135, 2020.

[31] H. Song, A. Varava, O. Kravchenko, D. Kragic, M. Y. Wang, F. T. Pokorny, and K. Hang, "Herding by caging: a formation-based motion planning framework for guiding mobile agents," *Autonomous Robots*, vol. 45, no. 5, pp. 613–631, 2021.

[32] Y. Deng, A. Li, M. Ogura, and N. Wakamiya, “Collision-free shepherding control of a single target within a swarm,” in *2023 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 5218–5223, 2023.

[33] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, “Shepherding behaviors with multiple shepherds,” in *2005 IEEE International Conference on Robotics and Automation*, pp. 3402–3407, 2005.

[34] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. 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, p. 20140719, 2014.

[35] Y. Chen and T. Kolokolnikov, “A minimal model of predator–swarm interactions,” *Journal of the Royal Society Interface*, vol. 11, no. 94, p. 20131208, 2014.

[36] W. Lee and D. Kim, “Autonomous shepherding behaviors of multiple target steering robots,” *Sensors*, vol. 17, no. 12, p. 2729, 2017.

[37] Y. Tsunoda, Y. Sueoka, Y. Sato, and K. Osuka, “Analysis of local-camera-based shepherding navigation,” *Advanced Robotics*, vol. 32, no. 23, pp. 1217–1228, 2018.

[38] 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. 214658–214671, 2020.

[39] 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*, pp. 2194–2201, 2020.

[40] J. Zhi and J.-M. Lien, “Learning to herd agents amongst obstacles: Training robust shepherding behaviors using deep reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4163–4168, 2021.

[41] A. Hussein, E. Petraki, S. Elsawah, and H. A. Abbass, “Autonomous swarm shepherding using curriculum-based reinforcement learning,” in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 633–641, 2022.

[42] A. Li, M. Ogura, and N. Wakamiya, “Communication-free shepherding navigation with multiple steering agents,” *Frontiers in Control Engineering*, vol. 4, p. 989232, 2023.

[43] A. Li, M. Ogura, and N. Wakamiya, “Proposal of a bearing-only shepherding algorithm with limited sensing capabilities,” in *AROB-ISBC-SWARM 2023*, pp. GS34–1, 2023.

[44] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros, “Mutual interactions, potentials, and individual distance in a social aggregation,” *Journal of Mathematical Biology*, vol. 47, pp. 353–389, 2003.

[45] A. Özdemir, M. Gauci, and R. Groß, “Shepherding with robots that do not compute,” in *Artificial Life Conference Proceedings*, pp. 332–339, 2017.

[46] T. Krajník, J. Faigl, V. Vonásek, K. Košnar, M. Kulich, and L. Přeučil, “Simple yet stable bearing-only navigation,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[47] R. Zheng, Y. Liu, and D. Sun, “Enclosing a target by nonholonomic mobile robots with bearing-only measurements,” *Automatica*, vol. 53, pp. 400–407, 2015.

[48] S. Zhao, Z. Li, and Z. Ding, "Bearing-only formation tracking control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4541–4554, 2019.

[49] A. J. Hepworth, A. Hussein, D. J. Reid, and H. A. Abbass, "Swarm analytics: Designing information markers to characterise swarm systems in shepherding contexts," *Adaptive behavior*, vol. 31, no. 4, pp. 323–349, 2023.

[50] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, pp. 513–516, 2005.

[51] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, p. 1226, 1995.

[52] E. A. Codling, M. J. Plank, and S. Benhamou, "Random walk models in biology," *Journal of the Royal Society Interface*, vol. 5, no. 25, pp. 813–834, 2008.