

# AGENT: an adaptive grouping and entrapping method for flocking systems

Chen Wang, Wenxi Kuang, Minqiang Gu and Zhun Fan\*

Shantou University, Shantou City, Guang Dong Province, 515063, China

\*Corresponding author. E-mail: [zfan@stu.edu.cn](mailto:zfan@stu.edu.cn)

## Abstract

This study proposes a distributed algorithm that enables agents' adaptive grouping and entrapment of multiple targets via automatic decision making, smooth flocking, and well-distributed entrapping. In this study, an agent distributed decision framework is proposed. Agents make their own decisions about which targets to surround based on environmental information. Meanwhile, a modified Vicsek model is proposed to enable agents to smoothly change formations to adapt to the environment, while forming an entrapping effect on the target. In addition, we provide an optional rotary entrapping function for this model to achieve better effect. We validate the performance of proposed method using simulation and physical experiments.

**Keywords:** swarm intelligence, robots, multi-agent systems, multiple targets entrapping, flocking, Vicsek model, distributed control

## Highlights:

- An adaptive decision-making model for swarm robots is proposed to entrap multiple targets.
- Swarm robots orbit the targets while entrapping them.
- Some indexes for evaluating the trapping performance were put forward.

## 1. Introduction

In recent years, the research of multi-agent systems has attracted extensive attention (Sheng et al., 2015; Zhu, 2015, 2020). Multi-agent systems can be divided into centralized, distributed, and hybrid ones in terms of decision-making methods they adopt. Many multi-agent systems were based on a centralized decision making (Loayza et al., 2017; Zhu, 2020). The centralized decision making has some advantages, such as centralized information processing and convenient management of data. However, they also have some significant drawbacks. For example, if the central information processing node fails, the whole system may collapse and stop working. Therefore, in recent years, many researches advocate decentralization and distributed decision-making methods that are gradually applied to more and more systems (Mullender, 1990; Gong et al., 2021). Compared with the centralized system, the distributed system has more difficulty in the coordination of agents, with information resources dispersed. However, it also has some notable and valuable advantages. Distributed systems have better scalability; that is, it is easy to add a node without affecting the functioning of other nodes in the system. At the same time, the distributed system has good robustness, which shows good resistance to damages of part of the system. There is also

a new paradigm that combines the characteristics of centralized and distributed systems, namely hybrid control architecture (Ye et al., 2011). There are many central control nodes that collectively influence the whole system's decision-making activities through communication. It is worth noting that this approach, while yielding better performance in some cases, is generally used for larger systems due to its complexity. The research discussed in this paper is mainly about distributed multi-agent systems.

There have been increasing research interests in the distributed cooperative control of multi-agent systems generating emergent flocking behaviors. These studies have received considerable attention since Reynolds proposed three heuristic rules (Reynolds, 1987) including collision avoidance, velocity matching, and flock centering for multiple agents. Based on the three general rules, hundreds of models have emerged to model the synchronized collective motions of animals, humans, or even migrating cells (Reynolds, 1987; Mastellone et al., 2007; Vicsek & Zafeiris, 2012; Fine & Shell, 2013; Mehes & Vicsek, 2014). Jinming Du proposed an evolutionary game theoretic approach to coordinated control of multi-agent systems. On the basis of the idea of natural selection, agents in the system act as the role of players in the game. Agents have different optional behaviors to choose as their strategies (Du, 2019). Jing et al., 2018 proposed two new distributed formation control schemes based on the weak rigidity theory under the condition of relying only on local relative position measurement (distributed and without communication). Jing et al., 2019 also proposed an angle-based distributed formation shape stabilization method for planar multi-agent systems. The angular rigidity theory is applied to the formation stabilization problem, and the formation with angle constraint is realized by multiple integral modeling agents. Jia & Vicsek, 2019 presented a general framework for modeling a wide selection of flocking scenarios under free boundary conditions. The stability of the model based on

Received: July 26, 2022. Revised: November 21, 2022. Accepted: December 15, 2022

© The Author(s) 2022. Published by Oxford University Press on behalf of the Society for Computational Design and Engineering. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact [journals.permissions@oup.com](mailto:journals.permissions@oup.com)

leader–follower relationship is studied when the collective motion is destroyed by random disturbances.

From the application point of view, applications of multi-agent systems include search and rescue (Baxter et al., 2007), area/border coverage (Rubenstein et al., 2014), deployment of sensor networks (Kim et al., 2014), collective transportation and construction (Rubenstein et al., 2014), and convoy/escorting missions (Antonelli et al., 2007; Barnes et al., 2009). Among them, target entrapping is a typical challenging research field (Zhang et al., 2018), for which much work has been done. Yang et al., 2020 proposed an estimator–controller framework for the robot. Within this framework, the robot can entrap the target without any prior position information in an arbitrarily shaped orbit. Yao et al., 2019 proposed circumnavigation control algorithms enabling multiple robots to orbit a target that achieves entrapping effect. However, these methods do not consider how to deal with obstacles in the environment. In addition, there are many methods such as behavior-based control methods (Antonelli et al., 2007; Phung et al., 2018), virtual structure methods (Kawakami & Namerikawa, 2008; Sato & Maeda, 2010), leader–follower control methods (Yu et al., 2019; Yang et al., 2020), and biological heuristic methods (Jin et al., 2012; Peng et al., 2016). However, most of them do not explicitly consider the situations when there are multiple dynamic targets in the environment. To the best of our knowledge, only a handful of efforts have focused on multiple target entrapping. Kubo et al., 2013 proposed a swarm robot multitarget entrapping algorithm. However, the multiple targets are stationary. Yasuda et al., 2014 used swarm robots to entrap and transport multiple targets based on evolutionary artificial neural networks. However, there are no obstacles in the environment.

The challenge in achieving multitarget entrapment for a large swarm of robots is designing a common self-organized behavioral model for each individual robot, which enables the robots in the field to surround each target as evenly as possible through their own decisions. The gene regulatory network (GRN) method has achieved superior performance in previous research on entrapping. Inspired by the genetic and cellular mechanisms that control biological morphogenesis, Jin et al., 2012 proposed a hierarchical GRN method that enables agents to generate entrapping formation according to environmental changes. Peng et al., 2016 proposed an improved GRN for entrapping multiple dynamic targets in an environment containing obstacles. On this basis, Fan et al., 2019 used genetic programming to automatically generate optimal GRN structures according to the scene and realized a better entrapping effect of the multiple targets in a variety of complex scenes with obstacles. However, existing GRN methods do not explicitly deal with the physical constraints of robots. It is well known that ignoring these constraints can degrade the performance of the system in real-world physical implementations.

Vásárhelyi et al., 2018 presented a flocking model for real drones, and the experiments demonstrated that the induced swarm behavior remained stable under realistic conditions for large flock sizes. This study describes some basic principles and swarm model design methods that drones should follow to complete the autonomous swarm tasks. It is a critical work that achieved the world's first outdoor autonomous flocking flight of 30 drones. It completed the task of flocking with an agent-based approach, which originates from the Vicsek model. However, this study only provides the basic swarm model, and the Unmanned Aerial Vehicle (UAV) swarm does not perform comprehensive tasks. In this work, we improved it and designed an adaptive force allocation model to entrap multiple targets. Inspired by Vásárhelyi et al. (2018), this study proposes an adaptive grouping entrapping

method (AGENT) based on an improved Vicsek model that considers physical motion constraints. Compared with the previous swarm studies, this study proposes a distributed decision-making mechanism of swarm agents, and the formation of agents in the swarm is not fixed or predefined. The agents' decision makings and entrapping pattern formations are adaptive to the environmental changes. The main contributions of this study are as follows:

1. This study proposes a multitarget entrapping model framework combining an adaptive decision-making mechanism and an improved Vicsek model.
2. The improved Vicsek model proposed in this study enables the agents to emerge uniformly distributed entrapping of targets with strong robustness.
3. The adaptive decision-making mechanism facilitates evenly dividing the agents into groups to entrap multiple targets according to environmental information.
4. In this study, several evaluation indexes are established to evaluate the effect of multitarget entrapment. Both simulated and physical experiments are used to validate the proposed model, with the physical experiments conducted on the E-puck2 platform.

The remainder of this paper is organized as follows. In Section 2, we describe the design of the velocity controller. In Section 3, the adaptive decision-making method is introduced. In Section 4, we provide the comparative experiments to verify the feasibility of the proposed method. Real-world experiments on the E-puck2 robot platform are presented in Section 5. Finally, Section 6 concludes the paper.

## 2. Velocity Control Mechanism

In this section, we introduce the method of entrapping targets by agents, which include several factors to be considered. First, when agents perform tasks in a swarm, they should keep their distance from each other (Vásárhelyi et al., 2018). In other words, when they are too close, they should produce a mutually exclusive velocity term. Likewise, they should produce repulsion velocity terms when they are too close to the targets to avoid collisions. In addition, agents need to avoid obstacles in a timely manner. Second, to entrap the target, agents need to stay within a certain distance from the targets. Finally, in practical engineering applications, the aforementioned motions must consider the mobility of the robots, so the accelerations of them are limited.

### 2.1. Close to the target

When an agent obtains the information of the targets, its goal is to reach a point at a certain distance from one of the targets. Therefore, agents need to have a velocity term pointing toward the target point and a smooth decay of velocity as they approach the target.  $D(\cdot)$  as an ideal braking curve has a smooth velocity decay function in space, with constant acceleration at high velocity and an exponential approach in space at low velocity (Vásárhelyi et al., 2018). In function  $D(\cdot)$ ,  $r$  is the distance between an agent and the expected stopping point. The  $p$  gain determines the crossover point between the two phases of deceleration, and  $a$  is the preferred acceleration.

$$D(r, a, p) = \begin{cases} 0 & \text{if } r \leq 0 \\ rp & \text{if } 0 < rp \leq a/p \\ \sqrt{2ar - a^2/p^2} & \text{otherwise} \end{cases} \quad (1)$$

With this smooth decay curve, agents can implement a velocity decay when approaching the target in equation (2). This is essentially what one does when pressing the brake pedal of a car. First, the brake pedal is pushed at a high velocity, and then the velocity is gradually decreased.

$$\mathbf{v}_{it} = [v_f + C^t \cdot D(r_{it} - R_{entrap}, a^t, p^t)] \cdot \mathbf{r}_{it} \quad (2)$$

In the above equations, we set the initial velocity  $v_f$  for all agents.  $C^t$  is the preferred common travelling velocity coefficient for all agents approaching targets.  $r_i$  and  $r_t$  represent the absolute position of agent  $i$  and its target, respectively.  $r_{it} = |\mathbf{r}_i - \mathbf{r}_t|$  is the distance between agents  $i$  and the target.  $a^t$  is the maximum acceleration allowed. Higher values assume that agents can brake quicker. Excessively high values result in the inability of agents to react to excessively large velocity differences in time and thus lead to collisions.  $p^t$  is the gain of the optimal braking curve used to determine the maximum allowed velocity difference. Large values approximate the braking curve to the constant acceleration curve. Small values elongate the final part of the braking (at a low velocity) with decreasing acceleration and smoother stops.  $\mathbf{r}_{it}$  is the direction in which the agent points to the target location.  $R_{entrap}$  represents the required distance of the stopping point in front of the target, which is predefined according to the entrapping task.

## 2.2. Repulsion

Agents generate velocity terms that move away from each other when the distance between agents is under  $r_{arep}$ , the distance at which the local repulsion kicks in. Larger values create sparser flocks with fewer collisions.

$$\mathbf{v}_{ij}^{rep} = \begin{cases} p_a^{rep} (r_{arep} - r_{ij}) \cdot \mathbf{r}_{ij} & \text{if } (r_{ij} < r_{arep}), \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $r_i$  and  $r_j$  represent the absolute position of agent  $i$  and agent  $j$ , respectively.  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$  is the distance between agents  $i$  and  $j$ .  $p_a^{rep}$  is the linear coefficient of the velocity of repulsion between agents, and  $\mathbf{r}_{ij}$  represents the direction of the velocity of acting on agent  $i$  from agent  $j$  to agent  $i$ .

Furthermore, repulsion is also used between the agents and targets unidirectionally. If the distance between the agent and the target is under  $r_{trep}$ , then the agents will be pushed away from the target.

$$\mathbf{v}_{itarget}^{rep} = \begin{cases} p_t^{rep} (r_{trep} - r_{it}) \cdot \mathbf{r}_{it} & \text{if } (r_{it} < r_{trep}) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Similarly,  $r_{it} = |\mathbf{r}_i - \mathbf{r}_t|$  is the distance between agents  $i$  and the target.  $p_t^{rep}$  is the linear coefficient of the repulsion velocity between the agent and the targets.  $\mathbf{r}_{it}$  represents the repulsion direction acting on agent  $i$  from the target. Each agent needs to calculate the repulsion velocity term for all targets.

To obtain aggregated repulsion, we take the vectorial sum of the interaction terms of repulsion introduced in equations (3) and (4):

$$\mathbf{v}_i^{rep} = \sum_{j \neq i} \mathbf{v}_{ij}^{rep} + \sum_{target} \mathbf{v}_{itarget}^{rep} \quad (5)$$

## 2.3. Interaction with walls and obstacles

In some practical applications, the task of entrapping targets by agents needs to be carried out within a certain area. In this study,

the flocking motion mechanism of the AGENT method considers the constraints of boundary (Yates et al., 2009; Tarcai et al., 2011) and obstacles of the arena. The targets and agents move in a square arena with walls and obstacles. To better avoid collisions with the walls, it is assumed that there are virtual agents distributed on the boundary of walls and obstacles. The virtual agent is located at the point closest to the agent on the boundary of the wall or obstacle (Han et al., 2006).

$$\mathbf{v}_{id}^{wall} = \begin{cases} 0 & \text{if } (r_{id} \geq r_{wall}) \\ C^d \cdot (v_{id} - D(r_{id} - r_{wall}, a^d, p^d)) \cdot \mathbf{v}_{id} & \text{otherwise} \end{cases} \quad (6)$$

where  $C^d$  is the velocity coefficient, and  $r_{wall}$  is the safe distance from the agents to the wall;  $r_d$  represents the absolute position of agent  $i$ 's closest point on the boundary of the wall or obstacle (virtual agent's position).  $r_{id} = |\mathbf{r}_i - \mathbf{r}_d|$  is the distance between agents  $i$  and the closest point on the boundary of the wall or obstacle (virtual agent's position).  $a^d$  and  $p^d$  are same as  $a^t$  and  $p^t$  in equation (2) but for avoiding collisions with walls.  $v_{id}$  is virtual agent's velocity that is perpendicular to the wall edge pointing inwards in the arena.  $\mathbf{v}_{id}$  ( $\mathbf{v}_{id} = \frac{\mathbf{v}_i - \mathbf{v}_d}{v_{id}}$ ) represents the unit direction vector of the agent's obstacle avoidance direction that is calculated from the vector difference between virtual agent and agent velocities.

Walls and obstacles are treated similarly in agents' obstacle avoidance. Agents can use the same method to avoid obstacles or walls while entrapping the targets; that is, for each agent and obstacle, the velocity component  $\mathbf{v}_{id}^{obs}$  can be defined similarly to equation (6). Parameters such as the minimum distance between the expected agent and the wall  $r_{wall}$  can be modified according to the actual needs to be applied to obstacles ( $r_{obs}$ ).

## 2.4. Revolution

In military applications, while the agents emerge the basic entrapping effect on the target, if they can further orbit the target, a better entrapping performance can be achieved. Take the UAVs entrapping the target as an example, if the target suddenly hovers in the air, both the UAVs and the target will hover and thus remain relatively stationary. However, this will cause the UAVs to lose its initial speed, which is not conducive to entrapping task (considering that the target suddenly changes from static to accelerated escape movement at this time). In addition, when the UAVs revolve around the target, it is similar to patrolling while executing the entrapping task. This can make more efficient use of their installed sensors (for example, it can make the cameras have a wider field of vision), facilitating UAVs to find other targets and take rapid countermeasures. In equation (7),  $\mathbf{v}_{tan}$  represents the normal vector of the difference between the agent and the target position vector.  $v_r$  is the magnitude of revolution velocity expected for the agent.

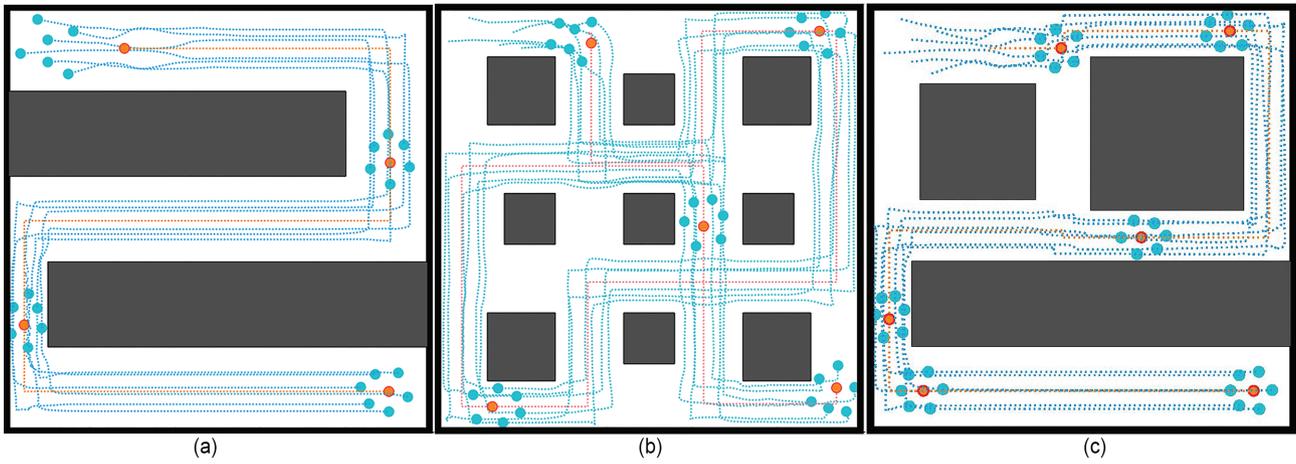
$$\mathbf{v}_i^{revolve} = v_r \cdot \mathbf{v}_{tan} \quad (7)$$

## 2.5. Final equation of desired velocity

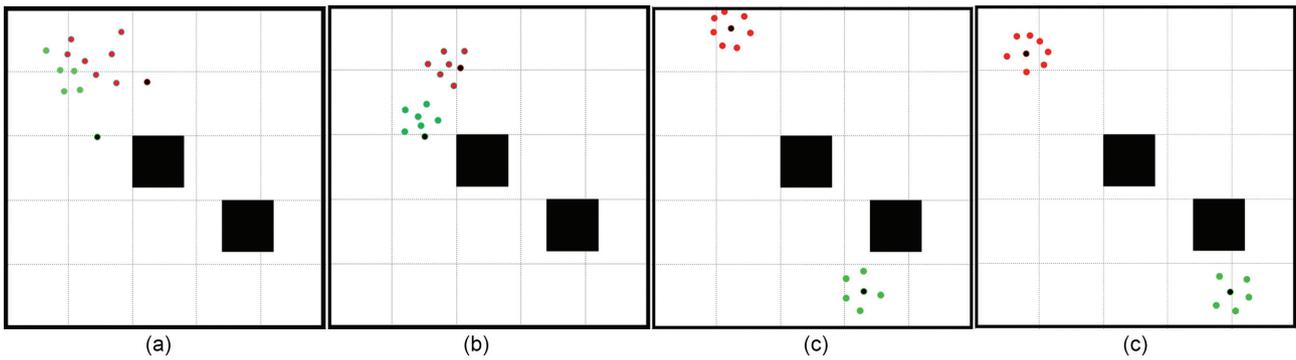
The speed controller needs to consider both of these possible effects, so the above velocity influence items need to be superimposed here. The desired velocity calculated by the algorithm is

$$\mathbf{v}_i^{desire} = \mathbf{v}_i^{rep} + \mathbf{v}_{it} + \mathbf{v}_{id}^{wall} + \mathbf{v}_{id}^{obs} \quad (8)$$

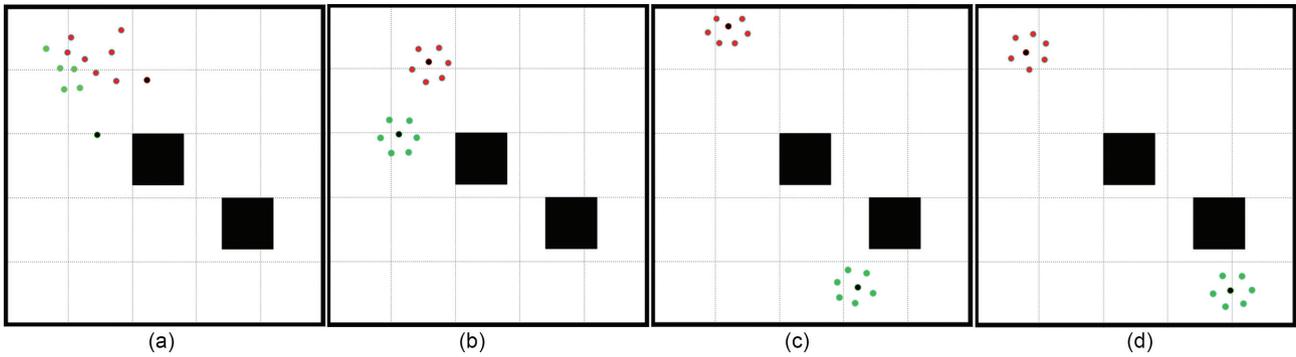
To make the method closer to the actual application, a velocity limit term  $v_{limit}$  is introduced. If the obtained velocity term is over the limit, its magnitude is reduced without changing the velocity



**Figure 1:** Process of agents entrapping the target in different scenes. (a) Scene 1. (b) Scene 2. (c) Scene 3.



**Figure 2:** Process of agents entrapping two targets with GRN. (a)  $t = 0$  s; (b)  $t = 14$  s; (c)  $t = 240$  s; and (d)  $t = 266$  s.



**Figure 3:** Process of agents entrapping two targets with our method. (a)  $t = 0$  s; (b)  $t = 14$  s; (c)  $t = 240$  s; and (d)  $t = 266$  s.

**Table 1:** Parameter setting of the AGENT model in simulation experiment.

Parameter	$a$	$b$	$C^t$	$R_{\text{entrap}}$	$a^t$	$p^t$	$p_a^{\text{rep}}$	$r_{\text{arep}}$
Value	0.22	3.2	2.6	8	5	4	2.55	18
Parameter	$p_t^{\text{rep}}$	$r_{\text{trep}}$	$C^d$	$\mathbf{v}_d$	$a^d$	$p^d$	$r_{\text{wall}}$	$r_{\text{obs}}$
Value	3.4	18	3.2	13.6	4	5	1.5	1.5

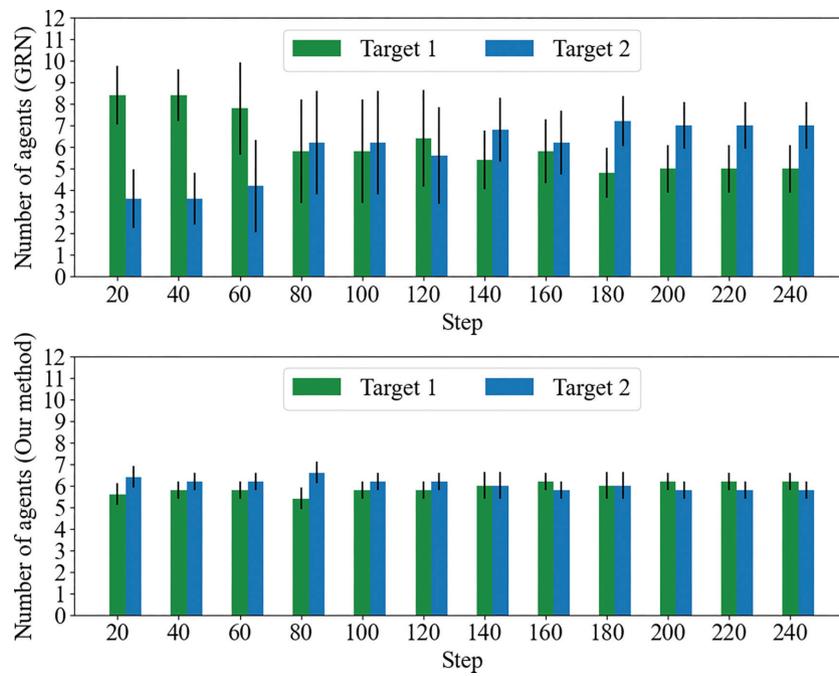
direction.

$$\mathbf{v}_i^{\text{desire}} = \frac{\mathbf{v}_i^{\text{desire}}}{|\mathbf{v}_i^{\text{desire}}|} \cdot \min \left\{ |\mathbf{v}_i^{\text{desire}}|, v_{\text{limit}} \right\} \quad (9)$$

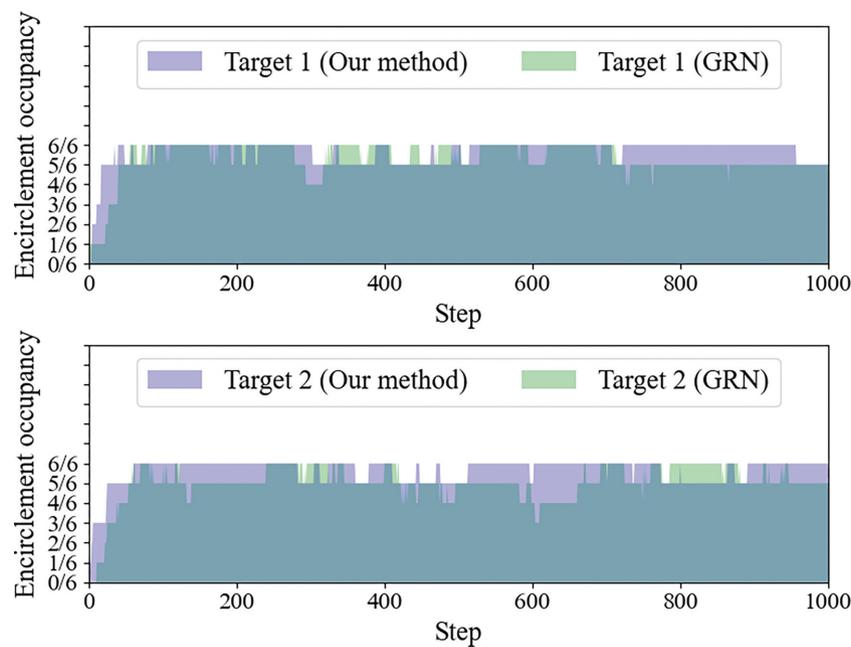
### 3. Adaptive Decision Making

During an entrapping mission that encounters multiple targets with equal significance, it is preferred that the agents are grouped evenly to encircle each target. In distributed systems, agents need to make decisions to surround corresponding targets, and the phenomenon of entrapping appears at the swarm level (van Veen et al., 2020). In the AGENT method, the problem of target grouping is transformed into the problem of agents selecting targets according to environmental factors.

In the task of entrapping multiple targets, the environmental factors to be considered include the number of agents surrounding the target and the relative distance between the agent and the target. If the agent is too far from the target A relative to the other



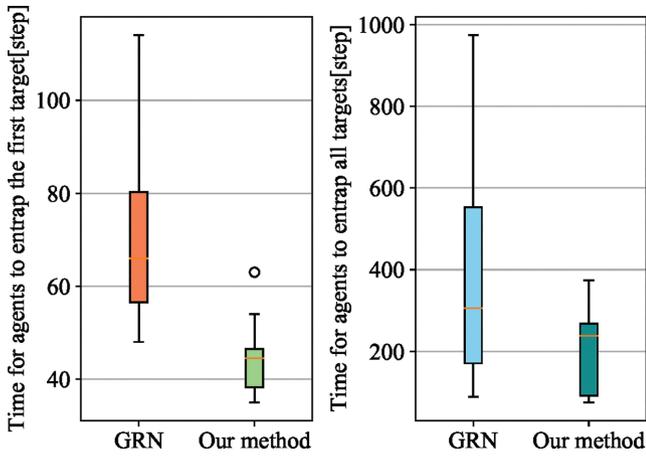
**Figure 4:** Number statistics of agents entrapping two targets. This is the statistical result of six simulation experiments (240 steps in every simulation experiment) with twelve agents and two targets in the same arena. Ideally, six agents should be assigned to each target. The figure shows the number of agents around the two targets at different sampling moments with the GRN and our AGENT method.



**Figure 5:** Statistics of the distribution of agents around the target. Twelve robots and two targets moved 1000 steps in the arena using the GRN and our method. We divided a certain size (radius = 32 m) circle of each target into six uniform fan-shaped areas, and counted the distribution of agents in the fan-shaped area with a sampling interval of one step.

targets, or there are already enough agents surrounding the target A, the agent should not choose target A but should choose the other targets to entrap; consequently, the agent no longer needs to entrap the target. That is to say, the agent usually needs to combine these two factors including the distance from the target and the number of agents surrounding the target in entrapping scenes. Thus, agents should be endowed with the following mechanisms:

The agents calculate their distance from various targets in real time and detect the number of agents surrounding each target. All agents calculate the  $Seq$  matrix, as shown in equation (10). In this manner, agents make decisions to divide themselves into different groups to entrap different targets autonomously. The correlation factor is considered on the right-hand side of the equation, including the distances from the agent to each target  $r_{tm}$  and the



**Figure 6:** Time for agents to entrap one target and all targets for the first time. Twelve robots entrap two targets in the 250 m \* 250 m arena, using GRN and our method. The figure shows the time of agents occupying all fan-shaped areas around one target and all targets.

number of agents surrounding each target  $N_{itn}$  in real time.

$$(Seq_1 Seq_2 \dots Seq_n) = (a \ b) \cdot \begin{pmatrix} r_{it1} & r_{it2} & \dots & r_{itn} \\ N_{it1} & N_{it2} & \dots & N_{itn} \end{pmatrix}, \quad (10)$$

where  $(a, b)$  is the weight matrix that represents the importance of the two factors in the matrix. We can obtain a matrix representing the target entrapment sequences for each agent. Each agent only needs to entrap the target corresponding to the element sequence with the smallest  $Seq$  value in the  $Seq$  matrix. Furthermore, the  $Seq$  matrix is updated in time by each agent. Thus, agents make more suitable decisions for efficient entrapment.

We can continue to increase the parameters and corresponding weights to meet the actual needs such that different targets have different importance;  $c$  represents the weight of target's importance. For example, in equation (11),  $P_{itn}$  represents the different encirclement priorities of different targets. In this manner, the grouping algorithm in the AGENT method can be flexibly applied to a variety of scenes.

$$(Seq_1 Seq_2 \dots Seq_n) = (a \ b \ \dots \ c) \cdot \begin{pmatrix} r_{it1} & r_{it2} & \dots & r_{itn} \\ N_{it1} & N_{it2} & \dots & N_{itn} \\ \dots & \dots & \dots & \dots \\ P_{it1} & P_{it2} & \dots & P_{itn} \end{pmatrix} \quad (11)$$

As the decision-making framework shows, the factors and the weight of each factor that we need to consider in decision making can be increased according to the actual situation. This parameter matrix can be considered as the weight matrix of neural network. If there are many factors to consider, the method of deep neural network can be considered, which is just like classifying pictures according to pixel values. The dimensions of the weight matrix can be adjusted according to actual conditions. Therefore, the framework exhibits good migration and scalability. All the algorithms of the AGENT system have been introduced, with their pseudo-codes provided in Algorithm 1.

## 4. Simulation Experiment and Analysis

### 4.1. Simulation experiments

In this section, the performance of the proposed AGENT method is evaluated using simulation cases based on MATLAB. To demonstrate the validity and robustness of the AGENT method, we set

---

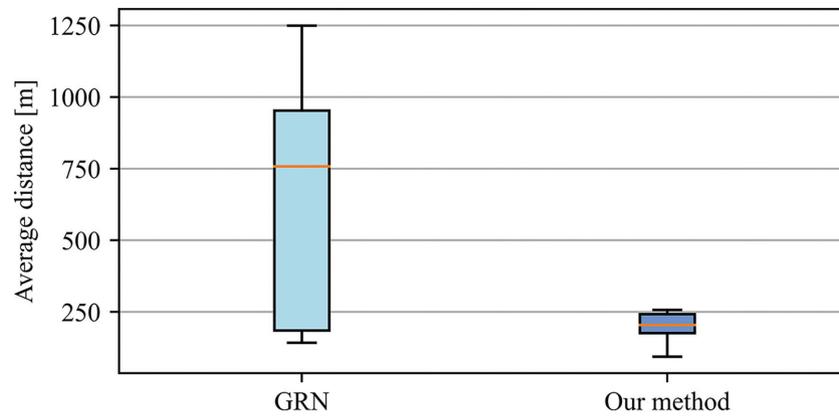
**Algorithm 1** Agent's adaptive decision on target selection and generation of target entrapment effect: pseudo-codes in AGENT system

---

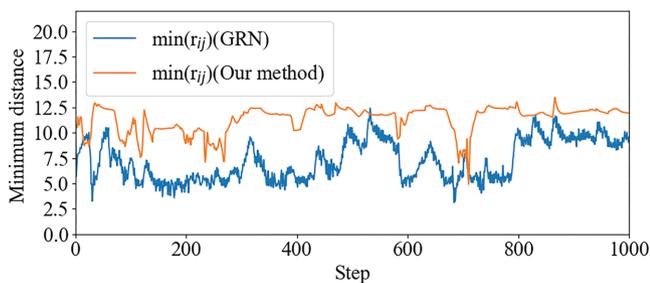
**Require:** The agent in swarm can obtain the position of agents in the swarm and the position of targets. The target is selected by the other agents in the swarm through communication or vision; the number of targets is  $n$ .

- 1: Initialize the environment parameters, including agent position, target position, obstacle position, etc.
  - 2: **while** (1) **do**
  - 3: **for**  $t$  **do** = 1 to  $n$
  - 4: Calculate the  $Seq$  value recording the distance of agent  $i$  from the target  $t$ , the number of agents that select the target  $t$  as goal.
  - 5: **end for**
  - 6: Agent  $i$  sorts the  $Seq$  values of each target and selects the target with the smallest  $Seq$  value as its target.
  - 7: Calculate the distance ( $r_{ij}$ ) between agent  $i$  and its neighbors in the swarm.
  - 8: Calculate the number of agents ( $k$ ) within  $r_{arep}$  distance of agent  $i$ .
  - 9: **for**  $j$  **do** = 1 to  $k$
  - 10: Calculate the repulsive influence velocity  $\mathbf{v}_{ij}^{rep}$  caused by neighboring agent  $j$  to agent  $i$ .
  - 11: **end for**
  - 12: Calculate the vector sum of the repulsive influence velocity of agent  $i$  from the above  $k$  neighbor agents.
  - 13: Calculate the distance ( $r_{it}$ ) between the agent  $i$  and targets.
  - 14: Calculate the number of targets ( $g$ ) within  $r_{trep}$  distance of agent  $i$ .
  - 15: **for**  $o$  **do** = 1 to  $g$
  - 16: Calculate the repulsive influence velocity  $\mathbf{v}_{itarget}^{rep}$  caused by target  $o$  to agent  $i$ .
  - 17: **end for**
  - 18: Calculate the vector sum of the repulsive influence velocity of agent  $i$  from the above  $g$  targets.
  - 19: Based on its distance from the chosen target, agent  $i$  generates the velocity  $\mathbf{v}_{it}$  of approaching the target and keeps a distance from its target  $R_{entrap}$  when it is too close.
  - 20: Calculate the distance ( $r_{id}$ ) between the agent  $i$  and the obstacle or boundary that the agent  $i$  needs to avoid to move within.
  - 21: Based on the distance between agent  $i$  and the nearest obstacle or boundary, the velocity of agent  $i$  to avoid the obstacle or boundary is calculated ( $\mathbf{v}_{id}^{wall}$  and  $\mathbf{v}_{id}^{obs}$ ).
  - 22: **if** The AGENT system enables the agent to orbit the target **then**
  - 23: Agent  $i$  calculates the velocity of revolution  $\mathbf{v}_i^{evolve}$  based on its position and its target position.
  - 24: **else**
  - 25:  $\mathbf{v}_i^{evolve} = 0$ .
  - 26: **end if**
  - 27: Agent  $i$  superimposes the above velocity influence terms on  $\mathbf{v}_i^{desire}$ . If the velocity amplitude exceeds  $v_{limit}$ , then the velocity amplitude is set to the maximum velocity  $v_{limit}$  of the agent without changing the velocity direction.
  - 28: **end while**
- 

different complex obstacles in a square scene, as shown in Fig. 1a, b, and c. The simulation experiment arena (250 m \* 250 m) was as follows: There were some agents in blue color and a target in orange color. The mission of the agents was to entrap the target and not crash into other agents, obstacles, or walls. The agents obtained the position information of each other through communication and detected the positions of obstacles and targets. The velocity of the target was 2.6 m/step and the velocity of the agents was 0–4 m/step. The trajectories of the target and agents are depicted in Fig. 1 along with pictures of the key moments when agents entrapped the target.



**Figure 7:** Average distance for agents to entrap all targets for the first time. Twelve robots entrap two targets in the 250 m \* 250 m arena, using GRN and our method. The figure shows the distance of agents occupying all fan-shaped areas around all targets.



**Figure 8:** Agents' minimum distance of entrapping process. Twelve robots entrap two targets in the 250 m \* 250 m arena, with GRN and our AGENT method. The figure shows the minimum distance of the agents during the 1000 steps.

In various complex obstacle scenes, agents can flexibly avoid obstacles and other agents, even in very narrow spaces (the space where agents are located has large obstacles or agents are very close to the obstacles). It can be seen from the trajectories of the agents that they avoid collision with obstacles in the arena during the entire entrapping process. Regardless of how the target changes direction, the agents can entrap the target with good performance.

To further demonstrate the adaptive decision ability of the AGENT method, we designed multiple target entrapping scenes, in which targets wandered in the arena. To make the agents and targets to move in a more realistic way, we used the Lévy flight model for target movement. The Lévy distribution is a probability distribution proposed by French mathematician Lévy in the 1930s (Viswanathan *et al.*, 2000), which is a random search path that obeys the Lévy distribution. This is a random walking mode that alternates between short- and long-distance searches, which conforms to the behavior trajectories of many natural creatures, such as bees and albatross.

The simulation experiment scene was as follows: There were red and green targets in the scene (250 m \* 250 m). The velocity of the target was 2.6 m/step, and the velocity of the agents was 0–4 m/step. The agent changed its color as it approached the target. The GRN method is famous for entrapping targets, which consists of two layers: the upper layer is for adaptive pattern generation, which is evolved by basic network motifs with genes and environmental inputs, and can generate a suitable pattern for entrapping the targets. The lower layer drives the robots to the target pattern generated by the upper layer. Fan *et al.*, 2019 further improved

the upper layer of GRN, in which genetic programming was used to automatically generate the optimal GRN upper layer according to the scene, and the experiments proved that the upper layer of GRN automatically optimized by genetic programming had better performance than the upper layer of GRN designed by human experts. To prove the superiority of our method, the AGENT and GRN methods are compared experimentally in Figs. 2 and 3. The simulation renderings of the agents entrapping two targets in the same scene are provided in the following figures and the video of the simulation experiments (its link is provided in the appendix). Also, the parameters involved in the AGENT model are listed in Table 1.

By comparing the pictures, both methods developed a formation that surrounded two targets to a certain extent. However, under the AGENT method, the number of agents was more uniform for entrapping each target in the arena, and the distribution of the agents' positions was more even, which demonstrates that the AGENT method was more capable of dealing with multiple targets than the GRN method in entrapping multiple targets.

## 4.2. Scalability analysis

To quantize the entrapping effect, we designed statistical indicators for the experiments. This study calculates the corresponding entrapping indicators in experiments to compare the entrapping effects of the two methods. First, we expect that the position distribution of the agents around the target should be as even as possible. In other words, the agents should not be too crowded in one direction of the target but should disperse as evenly as possible around the target. We define the occupancy rate of the encircling circle as the uniformity of the agent position around the target to evaluate the effect of entrapping. Then, in the mission of entrapping multiple targets in a swarm, we expect agents to be evenly divided into groups when the importance of the goal is the same. In this manner, multiple targets will be entrapped in an optimal way. In addition, the agents should entrap the target as soon as possible and with less traveled distance. Therefore, we calculate the response time and distance covered by the agents in entrapping targets. In addition, considering that the agents flock to perform tasks, we measured the minimum distance between the agents in the process of entrapping to measure the safety level of the two methods. Lastly, in practical applications, the motion of the swarm should be as stable as possible when performing tasks, to avoid frequently and suddenly changing directions. To meet the

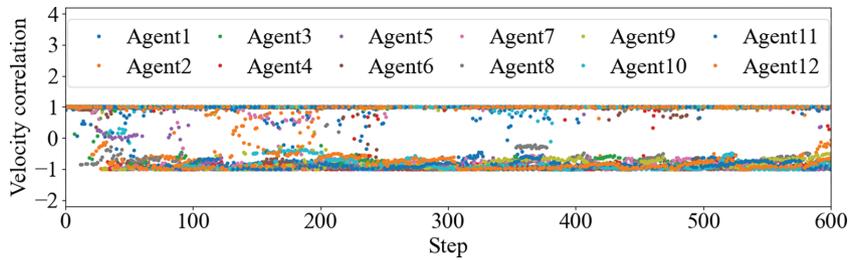


Figure 9: Velocity correlation of agents in the GRN method at continuous time (step).

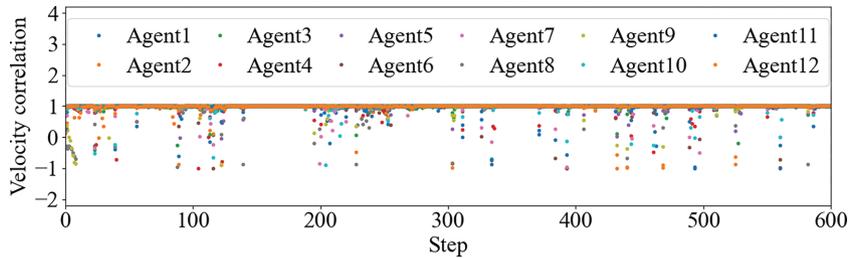


Figure 10: Velocity correlation of agents in the AGENT method at continuous time (step).

Table 2: Statistics of the time taken by robots to entrap two targets in six real-world experiments.

Experiments	1st	2nd	3rd	4th	5th	6th
Entrap one target	35 s	47 s	30 s	67 s	34 s	57 s
Entrap all targets	64 s	52 s	46 s	72 s	59 s	67 s

requirements of practical applications, we calculated the continuous velocity correlation of the agents in the entrapping process.

According to the above principles, six statistical indicators for evaluating the effectiveness of entrapping were designed as follows:

1. The number of agents entrapping each target.
2. The uniformity of the agents' positions around the target.
3. Time for agents to entrap the first target and all targets.
4. The average distance covered by agents to entrap all targets for the first time.
5. The minimum distance of agents during the entrapping process.
6. Velocity correlation of each agent's movement during the entrapping process.

We compare the AGENT and GRN methods using the above indicators. The statistical results are shown in Figs. 4–10. If six agents are assigned to each target, it is an ideal situation for the index of the uniformity of the number of agents entrapping the target. However, it is not an easy task for the agents to achieve that. The two targets wandered on the map with steps conforming to the Lévy distribution. If the agents' movements are not sufficiently flexible, they may not be able to form a timely, tight, and even encirclement of the dynamically moving targets over time. In the case of the same moving trajectory of targets, we calculated the indicator statistics of the methods in comparison on the implementation of the entrapping mission. First, we compare the number of agents allocated to each target by the AGENT and GRN methods. From Fig. 4, the AGENT method we propose has better

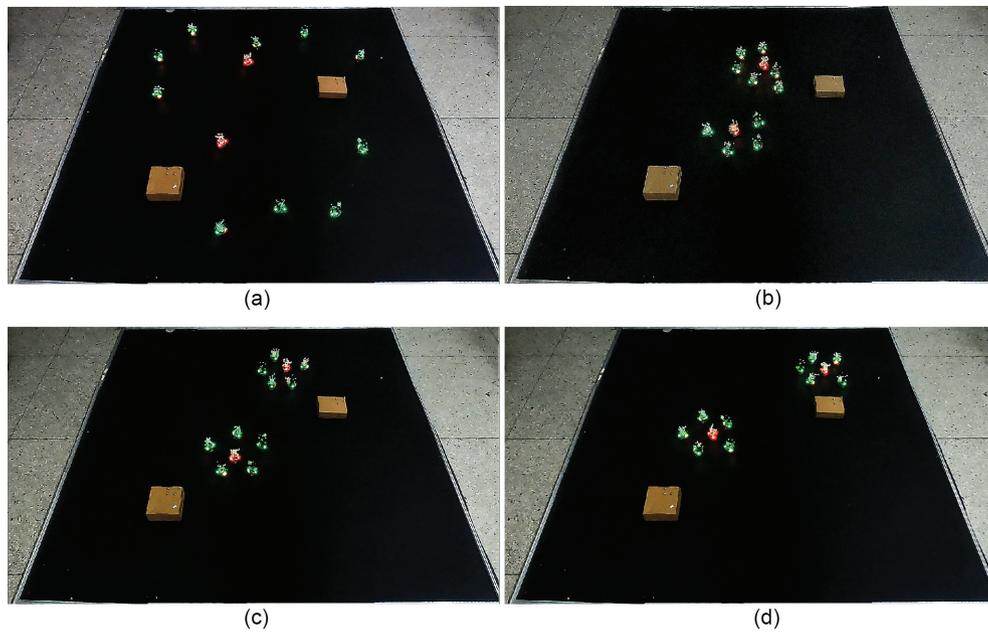
decision-making ability when entrapping multiple targets; that is, grouping is more even in numbers.

We then calculated the distribution of agents within a certain range in six directions around the target as the index of uniformity of the agents' positions around the target. In particular, an imaginary circle with the location of the target as the origin is setup with a predefined radius of 3.2 m in this study. We then divided this circle evenly into six sectors. To test whether the agents entrapped its target successfully, we counted the number of sectors with agents appearing in these sectors in each time step. From Fig. 5, in the AGENT method, the agents are more evenly distributed in the sectors, which shows that in this indicator the AGENT method is superior to the GRN method.

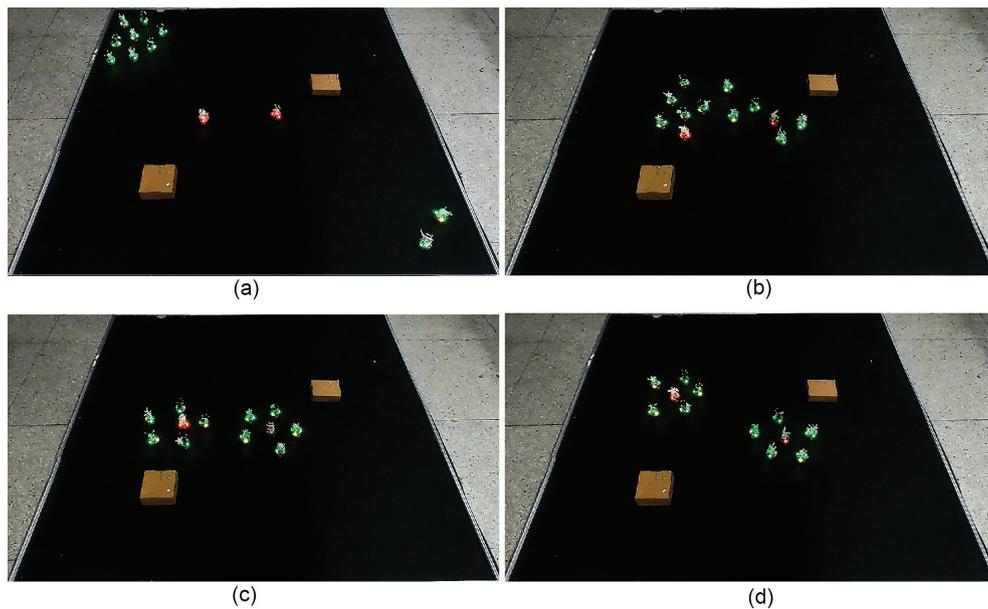
In this study, we assume that if the encirclement occupancy of the target is 6/6, then that means in each of the six sectors some agents appear to form the entrapment pattern, indicating that the target is successfully encircled. In a real entrap task, agents also need to achieve a uniform entrapping effect as quickly as possible. Therefore, this study counts the time that the agent first entered all the fan-shaped areas around the first target and all targets in Fig. 6. In addition, this study calculates the average distance of agents' first entrance in all fan-shaped areas around all targets, as shown in Fig. 7. It can be regarded as the average distance for agents entrapping all targets.

With the goal of providing more safety of practical applications of agents flocking to performing entrapping tasks, this study calculates some evaluation indicators related to the motion of flocking. First, we do not expect the distance between agents to become too small, which would be dangerous for agents while flocking in the real world. The average minimum distance was calculated as shown in Fig. 8. The agents should also make reasonable decisions to minimize sudden changes in the velocity direction, which jeopardizes the security of real-machine applications. The statistics are shown in Figs. 9 and 10, respectively. The formula for calculating the velocity correlation is as follows:

$$\phi^{\text{corr}} = \frac{v_i \cdot v_{i-1}}{|v_i| |v_{i-1}|},$$



**Figure 11:** Real-world experiments of agents entrapping two targets (scene 1). Ten E-puck2 robots entrap two targets in a 3 m \* 3 m arena. (a)  $t = 0$  s; (b)  $t = 28$  s; (c)  $t = 60$  s; and (d)  $t = 132$  s.



**Figure 12:** Real-world experiments of agents entrapping two targets (scene 2). Ten E-puck2 robots entrap two targets in a 3 m \* 3 m arena. (a)  $t = 0$  s; (b)  $t = 35$  s; (c)  $t = 67$  s; and (d)  $t = 139$  s.

where  $v_i$  represents the velocity of the agent at time (step)  $i$ , and  $v_{i-1}$  represents the velocity of the agent at the previous time (step) of  $v_i$ . As shown in Figs. 9 and 10, the angle between the current velocity of the agent and the velocity at the previous moment, the closer  $\phi^{\text{corr}}$  is to 1, and the more stable the velocity direction of the agent is. Conversely, the closer  $\phi^{\text{corr}}$  is to  $-1$ , the more drastically the velocity direction of the agent changes. We can clearly see that the agent using our AGENT method could obtain a more stable velocity direction than the GRN method.

When the agents use two methods to perform the entrapping task in the same scene, the AGENT method can ensure that the safe distance between agents is stable within an appropriate range, which is better than the GRN method as the

data in Fig. 8 show. The AGENT method with such a performance greatly improves the security during the flocking behavior. Furthermore, it can be seen from Fig. 9 that the agents may have some repeated jumps in the GRN (continuously changing the velocity direction by a large margin). This phenomenon is unfavorable for practical applications. It wastes a lot of movement resources due to decision errors and this movement is also dangerous for robots, especially in swarms. In our method, this sudden change in velocity direction is reduced (because the target will suddenly change direction, this behavior of agents should not be completely eliminated), as shown in Fig. 10.

The above six indicators prove the stability and superiority of the AGENT method from different directions; in general, the AGENT method developed in this study allows the agent system to achieve a good group entrapping effect.

## 5. Real-World Experiments

To perform real-world experiments on the AGENT method, we chose the E-puck2 robots to perform the entrapping task. In the arena with random obstacles and targets, 10 E-puck2 robots used the AGENT method to entrap two targets. The targets moved in the arena with the Lévy flight algorithm as their step generation mechanism. E-puck2 robot adopted differential wheel, and used the speed difference of two wheels to control the robot direction. The robots in the swarm communicated with each other via WiFi and obtained global information from the motion-capturing devices above the arena, including the position information of other robots, obstacles, and the boundary of the arena. The information was used for the robot to make decisions (regarding which target to entrap) and to make movement speed adjustments (to achieve an entrapping effect). If the robots entered all the sectors of the quintile circle within a certain radius of the target at the same time, the target was considered to be successfully encircled. We need to calculate how fast the swarm robot system can entrap the two targets, i.e., the shortest entrapping time for the swarm robot system to reach the enveloping circle with an occupancy rate of 100%. We counted the time (Table 2) taken by the E-puck2 robots to entrap the two targets in six experiments. From the table, we can conclude that E-puck2 robots can effectively and efficiently complete the task in less than 1.5 min. We selected two representative experiments, as shown in Figs. 11 and 12 (the real-world experiment video is also provided via the link of the appendix).

As shown in Figs. 11 and 12, in the case of the scattered distribution of the robots' positions, E-puck2 robots can adaptively group and entrap the targets evenly according to the environmental conditions, flexibly adjusting the formation to adapt to the environment, and avoiding to collide with neighbors and obstacles. Even if the target suddenly changes direction, the agent adaptively adjusts its speed to catch up with the target to form a tight circle. The real-world experiment verifies the effectiveness of the AGENT method.

## 6. Conclusions

This study proposes an adaptive grouping method to entrap multiple targets for distributed systems. Using our method, the agent can make decisions in real time based on environmental information, resulting in the effect of an even grouping and entrapping around the targets. The agents can flexibly respond to a sudden change in the direction of the target and always adapt to the change in movement speed to maintain the entrapping formation of the target. Furthermore, the flocking movement looks smooth and natural as possible, although environmental factors are complex and changeable. In conclusion, the AGENT method proposed by this research can be successfully applied and achieves superior performance in grouping and entrapping multiple targets in dynamic and complex environments.

## Acknowledgments

This research was supported in part by National Key R&D Program of China (2021ZD0111502), National Natural Science Foun-

dation of China (62176147), the Science and Technology Planning Project of Guangdong Province of China, the State Key Lab of Digital Manufacturing Equipment & Technology (DMETKF2019020), and by the National Defense Technology Innovation Special Zone Project(193-A14-226-01-01).

## Conflict of interest statement

There is no conflict of interest statement.

## References

- Antonelli, G., Arrichiello, F., & Chiaverini, S. (2007). The entrapment/escorting mission for a multi-robot system: Theory and experiments. In *2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*(pp. 1–6). <https://doi.org/10.1109/AIM.2007.4412504>.
- Barnes, L. E., Fields, M. A., & Valavanis, K. P. (2009). Swarm formation control utilizing elliptical surfaces and limiting functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **39**, 1434–1445. <https://doi.org/10.1109/TSMCB.2009.2018139>.
- Baxter, J. L., Burke, E., Garibaldi, J. M., & Norman, M. (2007). Multi-robot search and rescue: A potential field based approach. In *Autonomous robots and agents*(pp. 9–16). Springer. [https://doi.org/10.1007/978-3-540-73424-6\\_2](https://doi.org/10.1007/978-3-540-73424-6_2).
- Du, J. (2019). An evolutionary game coordinated control approach to division of labor in multi-agent systems. *IEEE Access*, **7**, 124295–124308. <https://doi.org/10.1109/ACCESS.2019.2938254>.
- Fan, Z., Wang, Z., Zhu, X., Hu, B., Zou, A., & Bao, D. (2019). An automatic design framework of swarm pattern formation based on multi-objective genetic programming. preprint arXiv:1910.14627. <https://doi.org/10.48550/arXiv.1910.14627>.
- Fine, B. T., & Shell, D. A. (2013). Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, **35**, 195–219. <https://doi.org/10.1007/s10514-013-9338-z>.
- Gong, Y., Zhang, B., & Xiong, M. (2021). Fully distributed dynamic event-triggered consensus control for multi-agent systems under dos attacks. In *2021 China Automation Congress (CAC)* (pp. 2698–2703). <https://doi.org/10.1109/CAC53003.2021.9728492>.
- Han, J., Li, M., & Guo, L. (2006). Soft control on collective behavior of a group of autonomous agents by a shill agent. *Journal of Systems Science and Complexity*, **19**, 54–62. <https://doi.org/10.1007/s11424-006-0054-z>.
- Jia, Y., & Vicsek, T. (2019). Modelling hierarchical flocking. *New Journal of Physics*, **21**, 093048. <https://doi.org/10.1088/1367-2630/ab428e>.
- Jin, Y., Guo, H., & Meng, Y. (2012). A hierarchical gene regulatory network for adaptive multirobot pattern formation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **42**, 805–816. <https://doi.org/10.1109/TSMCB.2011.2178021>.
- Jing, G., Zhang, G., Joseph Lee, H. W., & Wang, L. (2018). Weak rigidity theory and its application to formation stabilization. *SIAM Journal on Control and Optimization*, **56**, 2248–2273. <https://doi.org/10.1137/17M1122049>.
- Jing, G., Zhang, G., Lee, H. W. J., & Wang, L. (2019). Angle-based shape determination theory of planar graphs with application to formation stabilization. *Automatica*, **105**, 117–129. <https://doi.org/10.1016/j.automatica.2019.03.026>.
- Kawakami, H., & Namerikawa, T. (2008). Virtual structure based target-enclosing strategies for nonholonomic agents. In *2008 IEEE International Conference on Control Applications*(pp. 1043–1048). <https://doi.org/10.1109/CCA.2008.4629702>.

- Kim, S., Oh, H., Suk, J., & Tsourdos, A. (2014). Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Engineering Practice*, **29**, 42–49. <https://doi.org/10.1016/j.conengprac.2014.04.003>.
- Kubo, M., Sato, H., Yamaguchi, A., & Yoshimura, T. (2013). Target enclosure for multiple targets. In *Intelligent autonomous systems 12* (pp. 795–803). Springer. [https://doi.org/10.1007/978-3-642-33932-5\\_75](https://doi.org/10.1007/978-3-642-33932-5_75).
- Loayza, K., Lucas, P., & Peláez, E. (2017). A centralized control of movements using a collision avoidance algorithm for a swarm of autonomous agents. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)* (pp. 1–6). <https://doi.org/10.1109/ETCM.2017.8247496>.
- Mastellone, S., Stipanovic, D. M., & Spong, M. W. (2007). Remote formation control and collision avoidance for multi-agent nonholonomic systems. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation* (pp. 1062–1067). <https://doi.org/10.1109/ROBOT.2007.363125>.
- Mehes, E., & Vicsek, T. (2014). Collective motion of cells: From experiments to models. *Integrative Biology*, **6**, 831–854. <https://doi.org/10.1039/c4ib00115j>.
- Mullender, S. (1990). Distributed systems. ACM. <https://doi.org/10.1145/90417>.
- Peng, X., Zhang, S., & Lei, X. (2016). Multi-target trapping in constrained environments using gene regulatory network-based pattern formation. *International Journal of Advanced Robotic Systems*, **13**, 1729881416670152. <https://doi.org/10.1177/1729881416670152>.
- Phung, N., Kubo, M., Sato, H., & Iwanaga, S. (2018). Agreement algorithm using the trial and error method at the macrolevel. *Artificial Life and Robotics*, **23**, 564–570. <https://doi.org/10.1007/s10015-018-0489-z>.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 25–34). <https://doi.org/10.1145/37401.37406>.
- Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, **345**, 795–799. <https://doi.org/10.1126/science.1254295>.
- Sato, K., & Maeda, N. (2010). Target-enclosing strategies for multi-agent using adaptive control strategy. In *2010 IEEE International Conference on Control Applications* (pp. 1761–1766). <https://doi.org/10.1109/CCA.2010.5611117>.
- Sheng, Y., Zhu, H., Zhou, X., & Hu, W. (2015). Effective approaches to adaptive collaboration via dynamic role assignment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **46**, 76–92. <https://doi.org/10.1109/TSMC.2015.2423653>.
- Tarcai, N., Virágh, C., Ábel, D., Nagy, M., Várkonyi, P. L., Vásárhelyi, G., & Vicsek, T. (2011). Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock. *Journal of Statistical Mechanics: Theory and Experiment*, **2011**, P04010. <https://doi.org/10.1088/1742-5468/2011/04/P04010>.
- van Veen, D.-J., Kudesia, R. S., & Heinemann, H. R. (2020). An agent-based model of collective decision-making: How information sharing strategies scale with information overload. *IEEE Transactions on Computational Social Systems*, **7**, 751–767. <https://doi.org/10.1109/TCSS.2020.2986161>.
- Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., & Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. *Science Robotics*, **3**, eaat3536. <https://doi.org/10.1126/scirobotics.aat3536>.
- Vicsek, T., & Zafeiris, A. (2012). Collective motion. *Physics Reports*, **517**, 71–140. <https://doi.org/10.1016/j.physrep.2012.03.004>.
- Viswanathan, G., Afanasyev, V., Buldyrev, S. V., Havlin, S., Da Luz, M., Raposo, E., & Stanley, H. E. (2000). Lévy flights in random searches. *Physica A: Statistical Mechanics and Its Applications*, **282**, 1–12. [https://doi.org/10.1016/S0378-4371\(00\)00071-6](https://doi.org/10.1016/S0378-4371(00)00071-6).
- Yang, Z., Zhu, S., Chen, C., Feng, G., & Guan, X. (2020). Entrapping a target in an arbitrarily shaped orbit by a single robot using bearing measurements. *Automatica*, **113**, 108805. <https://doi.org/10.1016/j.automatica.2020.108805>.
- Yao, W., Lu, H., Zeng, Z., Xiao, J., & Zheng, Z. (2019). Distributed static and dynamic circumnavigation control with arbitrary spacings for a heterogeneous multi-robot system. *Journal of Intelligent & Robotic Systems*, **94**, 883–905. <https://doi.org/10.1007/s10846-018-0906-5>.
- Yasuda, T., Ohkura, K., Nomura, T., & Matsumura, Y. (2014). Evolutionary swarm robotics approach to a pursuit problem. In *2014 IEEE Symposium on Robotic Intelligence in Informationally Structured Space (RIIS)* (pp. 1–6). <https://doi.org/10.1109/RIIS.2014.7009182>.
- Yates, C. A., Erban, R., Escudero, C., Couzin, I. D., Buhl, J., Kevrekidis, I. G., Maini, P. K., & Sumpter, D. J. (2009). Inherent noise can facilitate coherence in collective swarm motion. *Proceedings of the National Academy of Sciences*, **106**, 5464–5469. <https://doi.org/10.1073/pnas.0811195106>.
- Ye, D., Zhang, M., & Sutanto, D. (2011). A hybrid multiagent framework with q-learning for power grid systems restoration. *IEEE Transactions on Power Systems*, **26**, 2434–2441. <https://doi.org/10.1109/TPWRS.2011.2157180>.
- Yu, X., Ma, J., Ding, N., & Zhang, A. (2019). Cooperative target enclosing control of multiple mobile robots subject to input disturbances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **51**, 3440–3449. <https://doi.org/10.1109/TSMC.2019.2926534>.
- Zhang, S., Liu, M., Lei, X., Huang, Y., & Zhang, F. (2018). Multi-target trapping with swarm robots based on pattern formation. *Robotics and Autonomous Systems*, **106**, 1–13. <https://doi.org/10.1016/j.robot.2018.04.008>.
- Zhu, H. (2015). Adaptive collaboration systems: Self-sustaining systems for optimal performance. *IEEE Systems, Man, and Cybernetics Magazine*, **1**, 8–15. <https://doi.org/10.1109/MSMC.2015.2460613>.
- Zhu, H. (2020). Agent categorization with group role assignment with constraints and simulated annealing. *IEEE Transactions on Computational Social Systems*, **7**, 1234–1245. <https://doi.org/10.1109/TCSS.2020.3006381>.

## Appendix 1

The experiment videos are as follows:

Simulation experiment with AGENT entrapping system: [https://www.bilibili.com/video/BV1sa411r72q?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV1sa411r72q?spm_id_from=333.999.0.0).

Simulation experiment with AGENT entrapping system (introduce revolution mechanism): [https://www.bilibili.com/video/BV173411J7Zs?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV173411J7Zs?spm_id_from=333.999.0.0).

Real-world experiment with AGENT entrapping system:

Scene 1: [https://www.bilibili.com/video/BV1T4y197E3?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV1T4y197E3?spm_id_from=333.999.0.0).

Scene 2: [https://www.bilibili.com/video/BV1XL41177DC?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV1XL41177DC?spm_id_from=333.999.0.0).