Toward target search approach of swarm robotics in limited communication environment based on robot chains with elimination mechanism

Yandong Luo, Jianwen Guo, Guoliang Ye, Yan Wang, Li Xie, Xiang Wang, Shaohui Zhang and Xiaohui Yan

Abstract
Swarm robotics refers to artificial swarm systems composed of a large number of autonomous mobile robots with relatively simple structures and functions. One of the basic problems of swarm robotics involves the target search process, which entails a cooperative search using limited perception and local interaction of robots under a self-organizing mechanism. When communication is limited, the connectivity of swarm robotics may decrease, leading to the failure of the target search task. This article describes a new target search method based on the robot chain model and the elimination mechanism. The proposed method allows the target search task to be completed efficiently and reliably while maintaining the connectivity of the robots. Experimental results show that the proposed algorithm offers better performance than conventional techniques in terms of search speed and success rate, and provides an effective method for solving the target search problem using swarm robotics in limited-communication environments.

Keywords
Swarm robotics, target search, robot chain, elimination mechanism, limited-communication environment

Date received: 26 August 2019; accepted: 26 February 2020

Topic: Mobile Robots and Multi-Robot Systems
Topic Editor: Guanghui Wen
Associate Editor: Haibo Du

Introduction
Swarm robotics is an important application in the field of multi-robot systems. With the advantages of being synergistic, distributed, robust, and operating in real time, there is a significant potential for swarm robotics to be applied in solving various problems because of their universality, effectiveness, optimality, reliability, and scalability. Research on swarm robotics covers aggregation, area coverage, target search, and cooperative handling. Target search refers to the task of multiple robots searching for a specified target in space. The use of swarm robotics to accomplish target search tasks is promising because of the robustness, flexibility, and scalability of robot swarms, especially in cases where there is no human participation or where human intervention is infeasible.

One of the primary problems of target search tasks in communication-restricted environments using swarm robotics is how to search for specific targets cooperatively.
without global communication and positioning capabilities while relying on short-range communication (fixed communication radius), simple motion control (forward, left turn, and right turn), and simple obstacle-avoidance functions (near the sensor). This problem poses the following challenges: (1) The robot generally cannot acquire or calculate its own position; (2) even if it can obtain its own position, the robot cannot broadcast its position information to other individuals in time because of the lack of a global communication system; and (3) if random walk search algorithms are used, the communication connectivity of a swarm system with only short-range communication capabilities will be greatly reduced and result in the failure of the search task. The communication connectivity of a swarm system refers to the communication ability between robots and other individuals, which requires the robots to be within the communication range of one another.

Social insects such as ants often deposit pheromones during their foraging behavior. These pheromones can be regarded as markers for guiding the foraging direction of individuals, who choose which direction to walk in according to the pheromone concentration. Similar to the pheromone mechanism, robot chains inspired by biological foraging behavior use the robots themselves as markers, whereby robots communicate with each other as a chain connection to induce other individuals to complete the search for the target. The robot chain ensures good swarm connectivity in limited-communication environments.

Consequently, we propose a target search method based on robot chains with an elimination mechanism. The proposed method improves the communication connectivity of swarm robotics to satisfy the application requirements in limited-communication environments. By introducing the elimination mechanism to the robot chain, the search direction of the swarm robot system is optimized. Moreover, we design two different robot motion mechanisms so that the proposed method can be applied in both bounded and unbounded environments. The proposed method is validated in simulation tests in such bounded and unbounded environments, and its performance is compared with that of the robot chain-based self-organizing search method and path formation method in a swarm (PFIAS). The test results show that the proposed method achieves higher efficiency and a better success rate than the PFIAS algorithm.

The remainder of this article is organized as follows. The second section discusses related work before the third section introduces the target search task model, defines the robot state, and explains the flow and realization of the control algorithm. The experimental parameters, results, and comparisons with the state of the art are presented in the fourth section, and the conclusions to this study are given in the fifth section.

Related work

Some of the more well-known swarm robotics research projects are as follows: (1) The swarm-bots project conducted at Brussels Free University in Belgium has developed a new type of autonomous mobile robot, named the S-bot. An evolutionary algorithm was applied to the design of the S-bot controller, and the cooperative behavior of a large number of swarm robotics was studied. (2) A robot project at Hughes Research Laboratory is based on virtual pheromones, combining pheromone logic with robot motion to achieve a variety of coordinated behaviors and achieve more complex tasks. (3) The I-Swarm project has designed and manufactured nearly 1000 groups of micro-robots to perform different group tasks under micro-scene conditions. (4) MIT used the Swarm Bot Robot to design a directed diffusion algorithm, enabling the robot to achieve rapid diffusion in a closed area while maintaining network connectivity. (5) The Kilobot, developed by Harvard University, allows thousands of robots to determine their own method of alignment without human interference or central control.

In recent years, several studies have applied algorithms to the problem of collaborative control technology for swarm robotics, such as the virtual structure approach, leader-following algorithm, artificial potential field method, and bio-heuristic algorithm. Among them, the research of swarm robotics based on a bio-heuristic algorithm refers to the extension of swarm intelligence algorithms such as particle swarm optimization and ant colony optimization, which is one of the most researched areas of swarm robotics.

The target search task studied in this article is also prevalent in the foraging behavior of organisms. The emerging cooperative method provides a more efficient means of accomplishing specified tasks through the development of intelligent behavior. Therefore, several studies have designed swarm robot control algorithms for the target search task by imitating the intelligent behavior of organisms. For example, Mayet et al. simulated pheromone-based ant colony foraging behavior using chemical sensors and alcohol placement. Inspired by the aggregation behavior of disk-based Nettlestalk bacteria, Schmickl et al. designed a small aggregation search algorithm for communication traffic. Meng et al. proposed an intelligent control algorithm for swarm robotics based on bacterial chemotaxis, which can achieve target search and pursuit. The main disadvantage of these approaches is that they typically need a large number of robots to cover the whole area before they can complete the connection between the target and the search starting point. However, the target search task may fail because of some decrease in robot connectivity.

Similar to pheromones, robotic chains respond to the environment through individuals. However, because swarm robotics lack a kind of substance acting on the
environment as a marker, they can only use other robots as the marker substance. The communication between robots then provides a link between the chains, meaning other individuals are attracted to complete the target search. Moreover, previous studies have shown that robot chains are an effective method whereby swarm robotics can complete foraging tasks. Based on the concept of robot chains, a distributed multi-robot search method has been designed by Nouyan et al. to search for a target through the links of the robot chain (PFIAS algorithm).\textsuperscript{11}

Method

Task model

We selected two test-beds to analyze our control algorithms, namely unbounded and bounded environments. In the unbounded environment, there are more severe challenges and stricter requirements for the connectivity between robots. This is because a robot’s perception range is small when compared to the infinite search area. Thus, once a robot loses communication with the other robots, it is difficult to resume connectivity using random algorithms. In the bounded environment, although the probability of each robot perceiving other robots is greatly increased by the limited search area, we test the performance of the control algorithms by increasing the number of obstacles.

The target search model is shown in Figure 1. A group of robots has to form a path between two objects—denoted as the nest and the food—in the unbounded and bounded environments. This path is called a branch of the robot chain, and is formed by connecting multiple robots at a fixed distance (unit length of the robot chain). The set distance must be less than the communication radius of the robot to ensure connectivity among multiple robots. In the process of searching, we choose two special fixed robots as the nest (defined as Robot_Nest) and the food (defined as Robot_Food), and then use the transformation of three different behavior states (Node, Explorer, and Lost_Chain) in the robots to realize the formation and adjustment of the robot chain. These three robot states will be defined in the “Robot model” section.

Initially, all robots are in the “Explorer” state and aggregated around the nest. They regard the nest as the source of the robot chain, and move to a position where the distance from the nest is greater than the unit length of the robot chain according the rules of motion (detailed in the “Robot model” section). Once they perceive that the distance between themselves and the nest satisfies the condition of forming the robot chain, they start to transit to the “Node” state and self-organize into chains. The robots in the “Node” state (defined as Robot_Node) act as trail markers and attract “Explorer” robots (defined as Robot_Explorer) through an “Extension Mechanism” (detailed in the “Attributes of robots” section). Neighboring robots in the robot chain have to be able to sense each other to ensure connectivity. As the robots have no knowledge about the position of the food, the structures are oriented in random directions. Moreover, in this self-organized search process, the robots also leave the robot chain and join it again at a different position, leading to a continuous exploration of the environment until the food is perceived. We call this principle the “energy elimination mechanism” (detailed in the “Extension mechanism” section). Finally, a path is formed between the nest and the food, as shown in Figure 1.

Robot model

For bounded and unbounded environment, we respectively use two types of robots: Kilobots\textsuperscript{16} and the marXbot.\textsuperscript{29} Kilobots are minimalistic robots widely employed in swarm robotics research, with very limited capabilities provided by a small range of sensors and actuators. The Kilobot moves on a flat surface through a pair of vibration motors that allow the robot to perform a slip-stick differential drive motion. The function of Kilobots is too simple to recognize and avoid obstacles. Therefore, we use the marXbot to test algorithm in bounded environment that include lots of obstacles or more complex environment. The marXbot is a ground-based robot that moves through a combination of wheels and tracks. Although the marXbot has various sensors and actuators, we just use the same function with Kilobots except in adding a proximity sensor.

Overall, the functional requirements of our algorithm for robots are as follows: (1) communication sensor of range $R$; (2) ranging sensor that detects the distances (but not the direction) to robots within the communication radius; (3) proximity sensor, which is only used for obstacle avoidance in the bounded environment; and (4) simple movements such as turning and straight walking.

Attributes of robots

Each robot is an independent execution unit with the following main attributes:
State. According to the different robot behaviors, we define the following three robot states:

1. Node: A robot in this state is the node of a robot chain and remains fixed.
2. Explorer: A robot in this state performs the search task according to the robot motion mechanism defined in the next subsection.
3. Lost_Chain: A robot in this state is isolated from the robot chains, and cannot communicate with robots in the Node state.

Gradient. The purpose of gradient formation is to create a long-range sense of distance across the swarm system, thus providing a rough measure of distance from the nest to the end of the robot chain. The robot acting as the nest emits a message with a fixed value of zero, indicating that it is the nest and the source of the gradient. All other robots in the Node state that receive this message will assume a gradient value of 1 and broadcast this message. All remaining robots traverse the messages of neighboring Robot_Nodes, obtain the minimum gradient value \( X \), and sets their own gradient value to \( X + 1 \). This value is broadcast to other Robot_Nodes, allowing them to update and calculate their own gradient values. The Robot_Node with the largest gradient value is the end of the robot chain. With the help of this gradient formation in the robot chain structure, Robot_Explorers can determine their direction of movement according to changes in gradient and whether they are at the end of the robot chain. In addition, the gradient formation indirectly enables continuous exploration with the length of the robot chain as the optimization condition.

Chain_ID. By defining an ID for each robot chain (Chain_ID), each Robot_Explorer can determine which robot chain they are approaching. According to this message, Robot_Explorers can determine whether to expand this robot chain or move far away from it. This is the foundation of the guidance mechanism for the search process in swarm robot systems. Its implementation is as follows. The ID of the robot in the Node state with a gradient value of 1 is defined as the Chain_ID. With the exception of robots with a gradient value of 1, each Robot_Node obtains the Chain_ID from information sent by all neighboring robots with a lower gradient value than itself. If a robot does not belong to a robot chain, its Chain_ID is 0 as shown in Figure 2.

Energy. According to the gradient formation, the robots can approximately determine the length of the robot chain. We assume that the energy of a robot chain flows from the nest to the end of the chain according to the normal distribution. A longer robot chain means that the robot at the end will have a lower energy value. The energy value of each robot can be calculated as

\[
f = \omega \times F(g; \mu, \sigma)
\]

where \( f \) is the energy value at the end of the robot chain, expressed by the duration of the robot in the Robot.Node state, \( g \) is the gradient of the Robot_Node, \( \omega \) is a positive weighting factor, and \( \mu, \sigma \) are the mean and variance of the gradient of the robot chain, respectively. In the experiments, \( \omega, \mu, \sigma \) were set to 100, 0, and 5, respectively. From the above equations, it can be shown that robots that are further away from the nest have lower energy values.

Motion. In this study, we designed two motion mechanisms: Edge-Following and Sigmoid Motion. Sigmoid-Motion based on gradient formation enables Robot_Explorers to move according to the guidance of the gradient. Although this improves the efficiency of the search task, guidance using the gradient increases the probability of collision through the intersection of various robots’ motion directions. Once a collision occurs, the Robot_Explorers must take obstacle avoidance action, and the direction of obstacle avoidance may not be consistent with the direction of guidance, which may lead to the loss of communication between the robot and the robot chain. The Edge-Following mechanism is a simple feedback controller that can adjust and control the distance between itself and the robot chain. This guarantees the communication connectivity of the swarm system, especially in the unbounded environment, although it lacks gradient guidance so the search efficiency decreases. In the “Experiments in a bounded environment” section, we report the results of detailed
simulation experiments and performance verification for the two mechanisms.

1. Edge-Following

Robot_Explorers continuously traverse and search for the nearest Robot_Node in the surrounding robot chain, and move around the Robot_Nodes in a counterclockwise direction at a distance of Orbit_R. Each Robot_Explorer senses and records the distance to the nearest Robot_Node. Every Robot_Explorer compares the current distance with the previous distance to determine whether closer to the Orbit_R. If they are closer to the Orbit_R than the last step that means the Robot_Explorer moving in a right direction, they will move straight. If not, they will change the direction until find the right direction. This part is presented with pseudocode, by Algorithm 1.

2. Sigmoid-Motion

In Sigmoid-Motion, a moving robot (Robot_Explorer) attempts to follow the robot chains by the gradient distribution of them to achieve a guided search by robots that have no direction- or position-sensing capabilities. This is done using a simple feedback controller and the sigmoid function for evaluating the current position of the moving robots as shown in equation (3). To be specific, as a component of robot chain, Robot_Node continuously calculate its own gradient value (in the “Gradient” section) and broadcast it. Any moving robot within the communication radius of one or more Robot_NODE can use this message to find the Robot_NODE with maximum or minimum gradients in their neighbors as the target nodes and determine its distance from them. The higher the gradient, the closer to the end of the robot chains. By contrast, it gets closer to the nest. All Robot_Explorers evaluate their current position according the distance between themselves and the target nodes as shown in equation (3) and obtain the movement direction indirectly. The curve between the distance to the target nodes and the evaluation value is shown on the left side of Figure 3

\[ f_{\text{sig}} = \omega \times \text{sig}(\Delta d, k) \]  

where

\[ \Delta d = \sum_{i=1}^{N} \frac{d_i - d_{\text{des}}}{N} \]  

\[ \text{Sig}(x, k) = \frac{1}{1 + e^{-kx}} \]

where \( k \) represents the slope of the sigmoid function, \( \omega \) is a weighting factor, \( N \) is the number of Robot_NODE with maximum or minimum gradients that attract the Robot_Explorer motion, \( d_{\text{des}} \) is the effective threshold of the sensing range for the robot, and \( d_i \) is the distance from the current source of attraction.

It is clear from the curve that the less the value is, the better the Robot_Explorer is located for the target nodes (closer to the nest or the end of robot chains). Thus, combined with the position evaluation equation and simple feedback controller, the Robot_Explorer guidance search can be realized. The feedback controller can be described as: if the time derivative of the objective function is less than zero \( (df/dt < 0) \), the robot keeps its previous movement direction, otherwise the robot moves randomly to find the right direction.

The process of Sigmoid-Motion can be described as: the Robot_Explorers traverse and sense Robot_Nodes in the vicinity (the Robot_Nodes connected to the black Robot_Explorer with dotted lines in Figure 3), and select the Robot_Nodes with the largest and smallest gradients in the sensing range as target nodes for their guided movement. By the Location evaluation equation and the feedback controller, the Robot_Explorers move toward the nest or the end of the robot chains.

The Sigmoid-Motion algorithm is illustrated in Figure 4.

The distance of Input-Oriented Target Robot_NODE is the...
distance between Robot_Explorer and the Robot_Node with maximum or minimum gradients in their neighbor. Pre_Evaluation_Value is the evaluated position value of the Robot_Explorer calculated in the previous operation cycle, Pre_ID is the ID of the target-guiding Robot_Node in the previous cycle, and MAX_Value is the initial value of the evaluated position.

Extension mechanism

The robot chain extension mechanism is mainly used to convert Robot_Explorers to Robot_Nodes. A flow diagram of the process is shown in Figure 5. The process determines whether the position of a Robot_Explorer is at the end of a robot chain, and whether the distance between the Robot_Explorer and the Robot_Node at the end of the chain satisfies the requirements.

Energy elimination mechanism

A flow diagram of the robot chain energy elimination mechanism is shown in Figure 6. The energy elimination mechanism of the robot chain acts on the robot at the end of the chain, and causes a progressive decrease in energy with every step. When the energy value is less than zero, the robot detaches from the robot chain and its state changes to Explorer. The detached robot then looks for other robot chains to extend.

Figure 4. Guided movement mode based on sigmoid model.

Figure 5. Extension mechanism of robot chains.

Figure 6. Flow diagram of energy elimination algorithm for robot chains.
According to the energy computation given in the “Robot model” section, the energy value decreases as we move along the robot chain away from the nest. Thus, a longer robot chain implies a shorter waiting time for other robots to join the robot chain structure. In fact, a longer robot chain indicates a lower probability that food exists in the current region, or that the structure of the robot chain is not optimized, which makes the robot chain unable to connect to the food along a shorter path. Therefore, this mechanism plays a vital role in the formation of the robot chain and the search direction of the swarm system. In addition, this mechanism can solve the deadlock problem in the bounded environment, which is a common problem in robot chain-based search algorithms. The deadlock problem can be described as follows: when the robot chain reaches the boundary of the search area, the robots accumulate at the boundary and cannot move because the algorithm cannot extend the chain. However, the energy elimination mechanism allows any robot nodes that reach the boundary to leave the robot chain and search for other directions away from the boundary of the area.

**Algorithm flow**

The algorithm developed in this study is illustrated in Figure 7, and the probabilistic finite state machine of the

---

**Algorithm 1. Edge-Following mechanism.**

```
pre_Distance ← MAX_DISTANCE // Pre.Distance will stores the previous closest distance to the Robot.Node at this earlier point.

loop
    current ← MAX_DISTANCE // current will store current distance to nearest neighbor
    for all neighbors n do
        if measured.distance(n) < current then
            current ← measured.distance(n)
        end if
    end for
    if current > Orbit.R then // desired Edge-Following distance
        if current > pre_Distance then // further away from desired distance
            Turn Left
        else
            Move Straight
        end if
    else // getting closer to desired distance
        if current < pre_Distance then
            Move Straight
        else
            Turn Right
        end if
    end if
    pre_Distance ← current
```

---

**Figure 7. Diagram showing search process of swarm robotics.**
individual robot behaviors is presented in Figure 8. Initially, all the robots are in the Explorer state and are clustered around Robot_Nest as shown in Figure 7(a). Based on the mode of motion and the extension mechanism, the robots continuously move outward and extend the robot chain as shown in Figure 7(b) and (c). Based on the gradient and energy mechanism, the robot chains exhibit increasing gradients and decreasing energy values from Robot_Nest to the end of the robot chains. The energy of Robot_Nest at one end of the robot chain gradually decreases until it reaches zero, and then the state changes from Node to Explorer as shown in Figure 7(d). Following the modes of motion, the Robot_Explorer moves toward other robot chains until it attaches to the end of a new robot chain that it has encountered. Because the robots in the swarm are all independent execution units, crowding and collisions frequently occur in the swarm system, causing individual robots in the Explorer state to lose communication with the robot chain. Thus, the robot state may change from Explorer to Lost_Chain as shown in Figure 7(b). To ensure connectivity of the swarm system in an unbounded environment, we force any Robot_Lost_Chain to remain in a fixed position, waiting for other robot chains to extend to its vicinity. Once some robot chain is around the Robot_Lost_Chain, the Robot_Lost_Chain is converted back to Robot_Explorer, allowing it to continue extending the robot chain, as shown in Figure 7(c). In a bounded environment, we take full advantage of the closed area and the obstacle-avoidance function of the robot to keep robots in the Lost_Chain state moving with a linear motion and avoiding obstacles until they reach the vicinity of a robot chain. Finally, when the end of a robot chain perceives the presence of Robot_Food, a robotic chain is formed between Robot_Nest and Robot_Food to complete the foraging task, as shown in Figure 7(e).

Experiments

Experiments in an unbounded environment

Experiments were conducted in an unbounded environment using the Kilomo simulation platform and Kilobot robots. The robots were only equipped with infrared receiving/transmitting sensors and local short-range communication and ranging functions. Figure 9 shows examples of the swarm robotic search process with $N = 9$ robots. Table 1 lists the parameter settings used for this experiment. The maximum allowable completion time was set to 500 s. If the food was not perceived and connected by robot chain during this time period, the test was terminated and the experiment considered a failure.
Difficulty test. The difficulty of the test varied by changing the distance between the nest and the food. This test allows us to analyze the relationship between the different distances and the time required to complete the task. Experiments were carried out using $N = 9$ swarm robots with $D$ set to $\{150, 155, 160, 165, 170, 175, 180\}$. For each $(N, D)$ combination, we conducted experiments with eight different target locations, giving a total of 56 tests. The experimental results are shown in Figure 10. It can be seen that, as the target distance increases, the completion time increases linearly. The search is accomplished by forming a robot chain. Thus, for a greater distance $D$, more robots are needed to form the robot chain, and the time required to construct the robot chain is longer.

Success rate test and contrast experiment. The success rate test was conducted with a swarm consisting of $N = 9$ robots and target distances of $D = \{200, 210, 220, 230, 240, 250, 260, 270\}$. For each $(N, D)$ combination, we conducted experiments with eight different target locations, giving a total of 64 tests. And, we record the number of tests that successfully completed the search task, and divide by the total number of tests to get the success rate. The test results are shown in the blue curve in Figure 11. The relationship between the success rate and the target distance is fitted with a fourth-order equation. Due to the limited number of robots and the inability of each individual to sense the direction of other robots, the length of the robot chain is limited and there is no guarantee that the robot chain will form a straight line. As a result, the search efficiency begins to decline and the robot chain approaches its limit of extension when $D$ exceeds a certain range. Figure 9 displays examples of the robot chain searching process near the limit of extension for $N = 9$ and $D = 270$ mm. In this case, all robots are joined to the same robot chain. Meanwhile, we also used the same experimental parameters to compare our algorithm with previous algorithm. The results are shown in Figure 11. It is clear that the algorithm proposed in this article greatly improves the range and success rate of the robotic target search task.

Experiments in a bounded environment

Experiments in a bounded environment were performed on an ARGoS simulation platform, and the results were compared to those given by the PFIAS algorithm. The parameter settings are presented in Table 2. The maximum allowable completion time was set to 1000 s at 10 steps/s, this gives a maximum of 10,000 steps. If the food was not perceived during this time period, the test was terminated.
and the experiment considered a failure. For each combination of parameters, the experiment was repeated 100 times.

**Comparison of control performance for different swarm models.**

The parameters used in the simulation experiment were the same as those used to test the PFIAS algorithm, where $L \times L = 5 \times 5 \text{ m}^2$, Unit_L_RC = 0.3 m, $N \in \{10, 15, 20\}$, and $D = 1.80 \text{ m}$.

Figure 12 shows the efficiency with which robot swarms of different sizes completed a target search task. The red box plots display the results of the Edge-Following mechanism, the green box plots display the results of the Sigmoid-Motion mechanism, and the blue box plots display the results of the PFIAS algorithm. The results in Figure 12 show that, in a bounded environment without obstacles, the completion times of the PFIAS algorithm and our control strategy decrease with increasing system scale, and the decrease is approximately linear. This shows that the search efficiency increases as the size of the system increases. Meanwhile, we calculated the number of the tests that successfully completed the search task to calculate the success rate for per box with 100 evaluations. For $N = 10, 15,$ and 20, the success rates of the two different motion control strategies proposed in this work are 83%, 92%, and 98% (Edge-Following) and 85%, 93%, and 97% (Sigmoid-Motion). The success rates of the PFIAS algorithm are 63%, 79%, and 85%, so our control algorithm with Edge-Following or Sigmoid-Motion achieves a higher search efficiency and success rate than the PFIAS algorithm for each group size.

**Motion mechanism test.** Figure 12 shows that, when the separation Orbit_R between nodes of a robot chain is 0.30 m, the success rates of the Sigmoid-Motion and Edge-Following motion control modes at $N = 10, 15,$ and 20 are very similar (83%, 92%, and 98% and 85%, 93%, and 97%, respectively). However, the area covered by the green box plots in Figure 12 is generally larger than that covered by the red box plots, and the distribution produced by Sigmoid-Motion is generally greater than that of the Edge-Following mechanism.

Therefore, we further analyzed this phenomenon and found that it occurred because the value of Unit_L_RC was too small. This made the motion space of robots executing the Sigmoid-Motion narrow, increasing the probability of collision and decreasing the efficiency. This is proved by a search experiment with a larger search area and a larger distance between robots in the robot chain, as shown in Figure 14. The experimental parameters were adjusted as follows: $L \times L = 8 \times 8 \text{ m}^2$, Orbit_R = 0.75 m, $D = 4 \text{ m}$, $N \in \{15, 20\}$, Unit_L_RC = 0.70 m, and $R = 1.5 \text{ m}$. The experimental data from “Comparison of control performance for different swarm models” section are also presented in Figure 13. It can be clearly seen that, when the distance between the robots forming the robot chain is shorter and the search area is smaller, the Edge-Following mechanism produces a higher search efficiency. However, as the value of Unit_L_RC increases, the Sigmoid-Motion mechanism becomes more efficient than Edge-Following. Moreover, the success rate of the robots under the Sigmoid-Motion control reaches 98% and 99%, whereas that of robots using the Edge-Following mechanism is only 79% and 81% over an area of $8 \times 8 \text{ m}^2$. These results suggest that when the robot interval in the robot chain and the search area are small, Edge-Following is better; in contrast, when the node interval of the robot chain is large, Sigmoid-Motion will provide better results.

**Table 2.** Parameter settings for experiments in bounded environment.

| Symbol   | Meaning                   | Unit       | Value               |
|----------|---------------------------|------------|---------------------|
| $L \times L$ | Search space            | m × m     | $5 \times 5, 8 \times 8$ |
| $R$     | Communication radius     | mm         | {0.6, 1.5}          |
| $R$     | Detection Radius         | m          | {0.6, 1.5}          |
| Orbit_R | Trajectory radius between Robot_Explorer and Robot_Node | m          | {0.3, 0.7}          |
| Unit_L_RC | Distance between robots in robot chain | m          | {0.3, 0.7}          |
| $D$     | Distance between         | m          | {1.0, 1.2, 1.4, ..., 3.0, 4.0} |
| $N$     | System size              | Count      | {5, 10, 15, 20}    |
| $O$     | Number of obstacles      | Count      | {0, 5, 10, ..., 30} |
| Speed   | Robot movement speed     | m/s        | 0.2                 |

**Figure 12.** The box-and-whisker-plot (Larsen et al.31) shows 100 evaluations per box of the completion time when changing the group size and the motion mechanism. See the caption of Figure 10 for an explanation of the box-and-whisker plots.
Difficulty test. For a system with a scale of \( N = 15 \), we analyzed and compared the search efficiency and success rate of tasks with \( D = \{1.0, 1.2, 1.4, \ldots, 3.0\} \). The results are shown in Figures 14 and 15. In Figure 15, the area covered by red box plots is generally smaller than the area covered by blue box plots, which means that the stability using our control strategy is generally better than that of the PFIAS algorithm. In addition, the red box plots are all lower than the blue box plots, indicating that our algorithm has a higher search efficiency. It is worth noting that, with \( D = 2.80 \text{ m} \), the distribution of the blue box plots suddenly moves downward. This does not indicate an improvement in search efficiency, but rather that the low success rate at \( D = 2.80 \text{ m} \) did not produce enough data for the box plots.

In Figure 15, we record the number of tests that successfully completed the search task, and divide by the total number of tests (100 observations per experiment) to get the success rate. The red polyline is the success rate of our control strategy and the blue polyline is the success rate of the PFIAS algorithm. Overall, the red line is above the blue line, indicating that our control strategy has a higher success rate.

Obstacle test. Figure 17 shows the search efficiency of our control strategy and of the PFIAS algorithm for different numbers of obstacles. In the figure, the red box plots and the green box plots indicate the proposed control strategy using the Edge-Following and Sigmoid-Motion mechanisms, respectively (see the “Robot model” section). The results show that our control strategies have a higher search efficiency than the PFIAS algorithm in obstacle-affected environments. Edge-Following produces a higher search efficiency than Sigmoid-Motion when there are relatively few obstacles in the search environment. The reasons for this phenomenon were explained in the “Motion
mechanism test” section. Compared with the other two algorithms, the Sigmoid-Motion mechanism achieves better search efficiency when the number of obstacles is higher, even though it requires significantly more time to complete the search task. This is because the increasing number of obstacles hinders the operation of the Edge-Following mechanism, which seeks to maintain a fixed distance from the robot chain. However, the gradient-based Sigmoid-Motion mechanism is more flexible in complex search environments. In the X-arena tests, the Edge-Following, Sigmoid-Motion, and PFIAS approaches achieved success rates of 67%, 92%, and 60%, respectively. Moreover, compared with the completion time in the X-arena, the data indicate that Sigmoid-Motion offers outstanding performance in complex environments. Edge-Following is suitable for more open search domains. The success rates of the R-arena tests are shown in Figure 18. Even though the motion control efficiency of Sigmoid-Motion is lower than that of Edge-Following in small task areas with fewer obstacles, the overall success rate of Sigmoid-Motion is higher than that of Edge-Following.

Conclusions and future work

In this study, we introduced an elimination mechanism in the robot chain target search, and designed a swarm robot chain target search method based on the elimination mechanism to meet the application requirements in a communication-limited environment. The proposed method is compared with the RC algorithm and the PFIAS
algorithm proposed earlier by the European Union. The results show that our control strategy has a higher search efficiency and a greater success rate, while at the same time requiring less from the robot sensors (no need to sense the direction of the robot). The entire search process is completely distributed, and the search is accomplished only through local communication and ranging between the robots. Future work should experiment with applications in actual robotic searches, and also consider a dynamic search environment because the current research is only limited to a static search environment.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Natural Science Foundation of China (61703102), the Department of Education of Guangdong in China (61703102), the Department of Education of Guangdong in China (61703102), the National Natural Science Foundation of China (61703102), the Department of Education of Guangdong in China (61703102), and the Industrial-academic-research cooperation demonstration base of DGUT-HengLi Town.

**ORCID iD**

Jianwen Guo https://orcid.org/0000-0002-3331-743X

**References**

1. Brooks M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell* 2013; 7(1): 1–41.

2. Brambilla M, Ferrante E, Birattari M, et al. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell* 2013; 7(1): 1–41.

3. Gasparri A, Priolo A, and Ulivi G. A swarm aggregation algorithm for multi-robot systems based on local interaction. In: 2012 IEEE international conference on control applications. Dubrovnik, 3–5 October 2012, pp. 1497–1502. IEEE.

4. Liu X and Tan Y. Adaptive potential fields model for solving distributed area coverage problem in swarm robotics. In: *International conference on swarm intelligence*, Fukuoka, Japan, 27 July–1 August, 2017, pp. 149–157. Cham: Springer.

5. Li J and Tan Y. A probabilistic finite state machine based strategy for multi-target search using swarm robotics. *Appl Soft Comput* 2019; 77: 467–483.

6. Hu J and Lanzon A. Cooperative control of innovative tri-rotor drones using robust feedback linearization. In: 2018 UKACC 12th international conference on control (CONTROL), Sheffield, 5–7 September 2018, pp. 347–352. IEEE.

7. Roosevelt N, Lii H, Nagpal R, et al. Multi-robot foraging for swarms of simple robots. Dissertations & Theses—Gradworks, 2011.

8. Deneubourg J L, Aron S, Goss S, et al. The self-organizing exploratory pattern of the argentine ant. *J Insect Behav* 1990; 3(2): 159–168.

9. Goss S and Jean-Louis D. Harvesting by a group of robots. In: *Proceedings of the First European Conference on Artificial Life*, 1992, pp. 195–204.

10. Luo Y, Guo J, Zeng Z, et al. Robot chain based self-organizing search method of swarm robotics. In: *International conference on intelligent computing*, 2018. Cham: Springer.

11. Nouyan S, Campo A, and Dorigo M. Path formation in a robot swarm. *Swarm Intell* 2008; 2(1): 1–23.

12. Trianni V, Nolfi S, and Dorigo M. Cooperative hole avoidance in a swarm-bot. *Robot Auto Syst* 2006; 54(2): 97–103.

13. David P and Mike D. Pheromone robotics. *Autonomous robots* 2001; 11: 319–324.

14. Seyfried J, Szymanski M, and Bender N. The I-SWARM project: intelligent small world autonomous robots for micro-manipulation. *Lecture Notes in Computer Science*, 2004, vol. 3342, pp. 70–83.

15. Autefage V, Chaumette S, and Magoni D. Distributed collaborative system for heterogeneous swarms of autonomous mobile robots. In: *International conference on protocol engineering*, Paris, 22–24 July 2015, pp. 1–7.

16. Rubenstein M, Ahler C, Hoff N, et al. Kilobot: a low cost robot with scalable operations designed for collective behaviors. *Robot Auto Syst* 2014; 62(7): 966–975.

17. Ren W and Beard R. Decentralized scheme for spacecraft formation flying via the virtual structure approach. *J Guid Control Dynam* 2004; 27: 73–82.

18. Chen J, Sun D, Yang J, et al. Leader–follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. *Int J Robot Res* 2010; 29: 727–747.

19. Hoshino S and Maki K. Safe and efficient motion planning of multiple mobile robots based on artificial potential for human behavior and robot congestion. *Adv Robot* 2015; 29: 1095–1109.

20. Masar M. A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In: *IEEE international conference on intelligent engineering systems*, San Jose, 19–21 June 2013, pp. 271–275.

21. Kennedy J. Particle swarm optimization. In: *Proceedings of ICNN’95—international conference on neural networks*, Perth, 27 November–1 December 1995.

22. Dorigo M, Maniezzo V, and Colomi A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man & Cybern Part B Cybern* 1996; 26(1): 29–41.

23. Couceiro MS, Vargas PA, Rui PR, et al. Benchmark of swarm robotics distributed techniques in a search task. *Robot Auto Syst* 2014; 62(2): 200–213.

24. Marjovi A and Marques L. Multi-robot olfactory search in structured environments. *Robot Auto Syst* 2011; 59(11): 867–881.

25. Mayet R, Roberz J, Schmickl T, et al. Antbots: a feasible visual emulation of pheromone trails for swarm robots. In:
26. Schmickl T, Möslenger C, Thenius R, et al. Individual adaptation allows collective path-finding in a robotic swarm. *Int J Fact Autom Robot Soft Comput* 2007; 4: 102–108.

27. Schmickl T, Thenius R, Moeslinger C, et al. Get in touch: cooperative decision making based on robot-to-robot collisions. *Auton Agent Multi-Agent Syst* 2009; 18(1): 133–155.

28. Meng Y, Guo H, and Jin Y. *A morphogenetic approach to flexible and robust shape formation for swarm robotic systems*. Amsterdam: North-Holland Publishing Co., 2013.

29. Michael B, Longchamp V, Magnenat S, et al. The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In: *2010 IEEE/RSJ international conference on intelligent robots and systems*, Taipei, 18–22 October 2010. IEEE.

30. Jansson F, Hartley M, Hinsch M, et al. Kilombo: a Kilobot simulator to enable effective research in swarm robotics. arXiv preprint arXiv:1511.04285, 2015.

31. Larsen Russell D. Box-and-whisker plots. *J of Chem Edu* 1985; 62(4): 302.

32. Pinciroli C, Trianni V, Grady RO, et al. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell* 2012; 6(4): 271–295.