

Article

A Bacterial Chemotaxis-Inspired Coordination Strategy for Coverage and Aggregation of Swarm Robots

Laihao Jiang, Hongwei Mo * and Peng Tian

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; lhjiangkyo@126.com (L.J.); tianpengheu@hrbeu.edu.cn (P.T.)

* Correspondence: honwei2004@126.com

Abstract: Many bio-inspired coordination strategies have been investigated for swarm robots. Bacterial chemotaxis exhibits a certain degree of intelligence, and has been developed some optimization algorithm for robot(s), e.g., bacterial foraging optimization algorithm (BFOA) and bacterial colony chemotaxis optimization algorithm (BCC). This paper proposes a bacterial chemotaxis-inspired coordination strategy (BCCS) of swarm robotic systems for coverage and aggregation. The coverage is the problem of finding a solution to uniformly deploy robots on a given bounded space. To solve this problem, this paper uses chaotic preprocessing to generate the initial positions of the robots. After the initialization, each robot calculates the area only covered by itself as the fitness function value. Then, each robot makes an action, running or rotating, depending on coordination strategy inspired bacterial chemotaxis. Moreover, we extend this solution and introduce a random factor to overcome aggregation, which is to guide robots to rendezvous at an unspecified point. The simulation results demonstrate the superior performance of the proposed coordination strategy in both success rate and an average number of iterations than other controllers.

Keywords: bio-inspired; bacterial chemotaxis; swarm robots; coordination strategy; distributed control



Citation: Jiang, L.; Mo, H.; Tian, P. A Bacterial Chemotaxis-Inspired Coordination Strategy for Coverage and Aggregation of Swarm Robots. *Appl. Sci.* **2021**, *11*, 1347. <https://doi.org/10.3390/app11031347>

Received: 22 December 2020

Accepted: 27 January 2021

Published: 2 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The adaption behavior of biological creatures provides a source of inspiration for the coordination of robots. Chemotaxis behavior in response to the presence of chemical concentration gradients occurs during foraging and has inspired a number of research and applications in robotic systems. Some examples include target search [1,2], exploration of unknown environments [3], mine detection [4], as well as environmental monitoring [5–7].

The simplest microorganism with chemotaxis behavior is a bacterium. In this paper, we focus on a simple bacterium called *Escherichia coli* (*E. coli*) and *E. coli*-inspired coordination strategies for robots. In foraging, *E. coli* performs a biased random walk, which is called bacterial chemotaxis [8]. Indeed, each bacterium moves to the highest (lowest) concentration location of the stimulus source based on the comparison between previous memory concentration and the current ones. Actually, it is likely that most of the time, the bacteria exhibit swarming behavior through interactions. Chemotaxis behavior of *E. coli* can serve as an inspiration source for coordination strategies of swarm robots. One of the potential advantages is that *E. coli*-inspired coordination strategy does not require the robot to have a complex physical structure to reduce the production cost, and it also contributes to the scalability of the swarm robotic system. The other advantage is that *E. coli* does not need the long-term memory function of a particle in PSO [9]. This means that the physical robot does not need a large storage unit. Another great potential advantage is the application in the medical field. Various microrobots have been developed in micro-assembly, micromanipulation, and drug delivery applications [10–12]. In general, the main control strategy for robots depends on the motion response of remotely controlled magnetic fields. Magnetotactic bacteria, which are controlled by magnetic fields, are used

to manipulate microparticles to assemble large structures [10]. Swarming microrobots have been applied in micromanipulation inspired behaviors of bacteria [11]. The collective behavior of micro-robotic swarms is obtained by magnetic interactions (attractive and repulsive) among individual agents [12]. Hence, applying bacteria-inspired coordination strategies to the macro and micro swarm robotic systems is of great significance.

In recent decades, existing chemotaxis-inspired control works have focused on a single robot mimicking chemotaxis behavior and applied it to source search tasks [13–18], where the chemotaxis-inspired control strategies are used to steer a single robot to the extrema of an unknown gradient field. Meanwhile, for applications of swarm robotic systems, some tasks, such as path planning, task allocation, are generally converted to optimization problems. The bacteria-inspired methods are served as optimization algorithms [19–22]. However, there is no direct mapping between individual robot and bacterium in bacteria-inspired optimization algorithms, because the algorithms involve operations, such as reproduction and elimination, which would be impractical to implement in a real robot. Furthermore, there is a lack of research on coordination mechanisms between robots inspired by bacteria.

Thus, we aim to propose a bacteria-inspired coordination strategy for a swarm robotic system inspired by the chemotaxis of *E. coli*, in which individual robot maps a bacterium. We consider two swarm robotic tasks: Coverage and aggregation. Coverage is to uniformly deploy robots on a given bounded space without central coordination [23,24], as is illustrated in Figure 1a. The potential practical applications include mapping of unknown environments, environment exploration, and autonomous surveillance systems. Aggregation is to guide a number of autonomous robots to rendezvous in a common place [24], as shown in Figure 1b, which is a basic behavior amongst biological creatures, such as cockroaches [25], bacteria [26]. To the best of our knowledge, such coordination strategy of swarm robotic system for flexible switching between coverage and aggregation is relatively less studied.

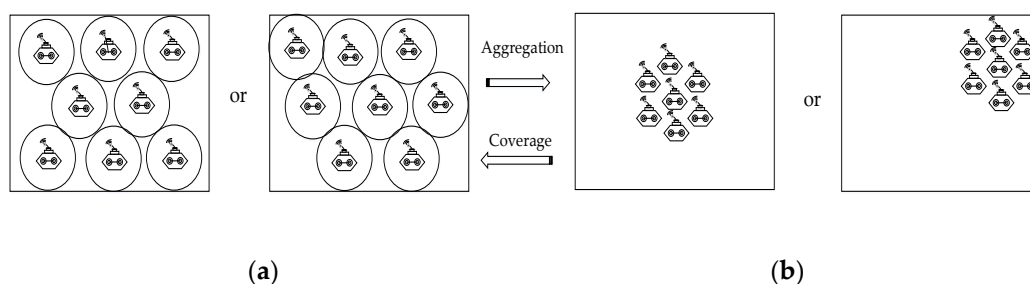


Figure 1. The coverage and aggregation: (a) Coverage; (b) aggregation.

The main contributions are as follows. (1) This paper deals with the coverage and aggregation using a bacterial chemotaxis-inspired coordination strategy named bacterial chemotaxis-inspired coordination strategy (BCCS), which is similar to the work in Reference [27]. The BCCS does not depend on the chemical stimulus and direct communication, just the position information of other robots within sense range. (2) The BCCS includes the chaotic preprocess that is to generate the initial positions of the robots in the beginning. This operation not only keeps the randomness of algorithm initialization, but also improves the diversity of positions of robots and the convenience of individual deployment. (3) In BCCS, the swarm robotic system is distributed, and the robots also do not need to be connected by a global network. Besides, the BCCS balances cohesive swarms and exploration of individuals to ensure that all robots rendezvous in common place under different robot numbers for aggregation. Finally, the simulation results show that the performance of BCCS and its variant is significantly better than that of the state-of-the-art chemotaxis controller.

The organization of this paper is as follows. Section 2 introduces bacteria inspiration related work. Section 3 describes problem statements and preliminaries. Section 4

presents the BCCS. The simulation results and discussion are provided in Section 5. Finally, conclusions and future work are drawn in Section 6.

2. Bacteria Inspiration

2.1. Bacteria Behavior

Bacteria exhibit three behaviors during foraging, which are simple and intelligent, as shown in Figure 2: Swimming, tumbling, and swarming. In the absence of chemical concentration, the bacteria perform a completely random motion. When subjected to a favorable change in concentration, bacteria will move straight for some time (swimming). The tumbling behavior means that the bacteria rotate randomly, and the tumbling frequency will increase when bacteria are subjected to an unfavorable change in concentration. Swarming emphasizes the interaction between bacteria. Bacteria secrete attracting or repelling chemicals that are similar to pheromone into the environment to communicate with each other and form the colony. In particular, these behaviors serve as inspiration for the robotic system.

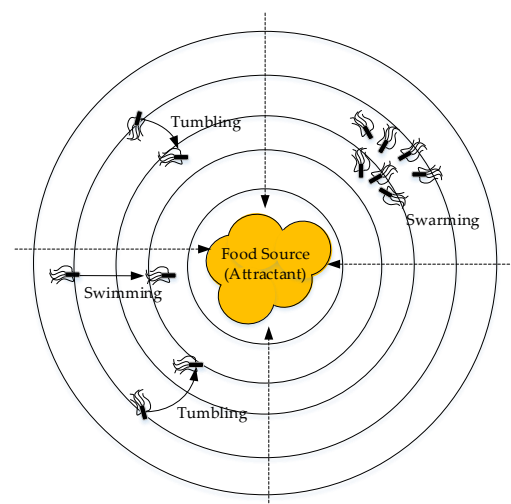


Figure 2. The basic behaviors of bacteria.

2.2. Applications

In the past few decades, the focus of robotic system has shifted to applying biology-inspired methods. The bacterial chemotaxis-inspired coordination control methods have been studied in a single robot and multi-robot system.

For a single robot, four reactive robot control algorithms based on chemotaxis are implemented and compared in Reference [17]. Among these algorithms, *E. coli*-based motion algorithm is the simplest and requires minimum structure and motion control requirements. However, it may only work with a very small system underground with steady changes of chemical concentrations. A Yuragi-based adaptive biased random walk strategy has been proposed to guide a mobile robot to locate a sound source in such an environment with and without gradient information [13]. This strategy can flexibly switch between the bacterial chemotaxis and Levy walk, which increases the success rate of the source search task. In Reference [14], a bacteria-inspired mobile robot, with a 1-DOF motor and a single sensor, has been proposed to navigate to a stimulus source, revealing the influence of environmental noise on navigation behavior. However, the related source searching algorithms based on bacterial chemotaxis cannot guarantee the repeatability of the path.

Related bio-inspired coordination strategies have received attention in swarm robots. Because the complexity of the system increases with the number of individuals, the coordination model should be simple, effective, and scalability. Bacteria swarming is a relatively simple biological system compared to insect swarms, bird flocks, and fish schools. Hence,

it is the natural source of swarm robotic system design. In Reference [28], two novel chemotaxis-inspired controllers were proposed for controlling the formation of robots. This paper quantifies the achieved degree of preset formation of robots as a global index and decomposes it into local indexes, which are used by each robot at each step. There is evidence that the bacterial chemotaxis-inspired strategy can guide robots to search multiple sources [29]. In this approach, the robots take advantage of gradient information to execute a simple motion strategy (biased a random walk) inspired by bacterial chemotaxis. This is more suitable for boundary coverage, and multiple dissipative sources search tasks. A bacteria-inspired coordinate model for agents with adaptable interactions is proposed, which includes long-range attraction, intermediate-range alignment, and short-range repulsion [30]. This leads to a collective motion similar to bacteria swarming. Especially, agents can adjust the peers' influence depending on the number of their neighbors. A novel bacterial chemotaxis model has been introduced, which improved the average escape rate by reducing the tumbling frequency on obstacle-laden terrains [31]. Related extend work, distributed gradient descent (DGD), has been used for mass scale evacuations in a dynamic environment [32]. In the evacuation process, each agent combines messages and computations in a simple vocabulary, and the neighborhood graph of each agent is dynamic changes. In addition to the above motion coordination models, bacteria can also serve as inspiration for task allocation [33]. The performance of the proposed task allocation method has been proved to be better than the three traditional methods in the net throughput and runtime.

Numerous researchers use bacteria-inspired methods for the coverage [5–7,34,35]. These methods mainly use robots to perform pollution visualization [5–7], and environmental exploration [34,35]. However, the robots need the concentration information, which is not available in our hypothetical environment. Moreover, the research of the bio-inspired coordination strategies considering two or more tasks is insufficient. In Reference [35], a coordination strategy of swarm robots inspired by bacterial chemotaxis behavior is proposed for target search and trapping. First, the map is preprocessed and divided into Voronoi cells. Each robot covers its Voronoi cell using bacterial chemotaxis (BC) algorithm, and then the whole area is covered by all the robots. After that, the robots respond to the stimulus from targets and trap them. The work in Reference [27] introduced a bacteria-inspired controller for coverage and rendezvous. The effectiveness of the controller is verified by simulations and physical experiments in a multi-robot system. However, the controller is more dependent on the initial distribution of robots and fails to analyze the schemes for the different numbers of robots in different size environments. Especially for aggregation, Reference [27] assumes all robots are connected by a graph G , which is unrealistic in a real swarm robotic system with numbers of individuals. Different from the work in Reference [27], our proposed coordination strategy just requires the local sense of the robot, and the system is distributed without a central controller and global network. The summary of related works is shown in Table 1.

Table 1. The summary of related works.

	Related Works	Coverage	Aggregation	Multitask
Related bacteria chemotaxis-inspired control strategies	[5]	✓		
	[6]	✓		
	[34]	✓		
	[35]	✓		✓
	[27]	✓	✓	✓
	This paper	✓	✓	✓

3. Problem Statements and Preliminary

3.1. Problem Statements

Our goal is to solve coverage and aggregation for swarm robots in light of the chemotaxis behavior of an individual bacterium. The coverage and aggregation in this paper are

that robots are used to maximize and minimize the covered area, respectively. We require a swarm robotic system with parallel processing capability and coordination mechanism between robots. So, we make the following assumptions to construct the schemes.

- (1) **Environment.** The environment $E \subset \mathbb{R}^2$ is unobstructed and bounded. The size denotes $L \times L$. L is the number of units for the length of horizontal (ordinate) range. There are no chemical signals in the environment.
- (2) **Robots.** The robot i is labeled as r_i , $i \in N$, $N = \{1, 2, \dots, n\}$. Each robot is omnidirectional and can move in any direction. The physical structure of the robot is ignored. The robot can only sense the presence of other robots within a sense radius. The robots can collaborate with each other only by using the position information of their neighbors.
- (3) **Radius.** As shown in Figure 3, we define R_s and R_i as the covered radius and sense radius, respectively, in which $R_i = 2R_s$. The robot can cover circular areas with a radius smaller than R_s . The robot can sense others within the R_i . Given robot i , the set of its neighbors is defined $N_i \triangleq \{s \in N \mid \|x_s - x_i\| \leq R_i, i \neq s\}$, and $x \in \mathbb{R}^2$ is the position of the robot. N_i follows the conditions: (i) $N_i \subset N$; (ii) $i \notin N_i$; (iii) $s \in N_i$.

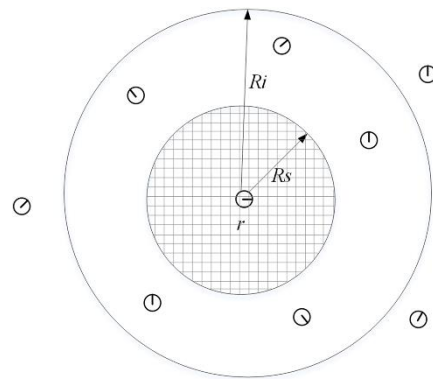


Figure 3. Covered radius R_s and sense radius R_i .

Next, we formulate the coordinate strategy for the swarm robotic system, as follow. Here, we consider a swarm of robots in a bounded region of 2D-space or 3D-space. The update of state for each robot is as follows:

$$x_i(t+1) = x_i(t) \oplus g\left(x_i(t), [x_j(t)]_{j \in N_i(t)}\right), \quad (1)$$

where $x_i(t) = (x_{i,1}(t), x_{i,2}(t), \dots, x_{i,n}(t))^T \in \mathbb{R}^n$ is the state variable of robot i at time t , $g(x_i(t), [x_j(t)]_{j \in N_i(t)})$ is a coordinate strategy that robot i updates its state variable according to the state variables of itself and its neighbors at time t . The symbol $\oplus$ denotes the strategy $g(x_i(t), [x_j(t)]_{j \in N_i(t)})$ is performed on the state variable of the $x_i(t)$.

3.2. Chaotic Model

Chaos phenomenon can be seen everywhere in nature, and it also widely exists in nature fields, such as bacteria, ants, and bees. Chaos motion is a kind of random behavior between rule and randomness in a completely deterministic system without external random factors. Chaos motion is sensitive to the initial value and chaotic parameter [36]. For the appropriate initial value and chaotic parameter, chaos motion has the ability to reach all states over a range of states without repeating. In view of the ergodicity of chaos motion, the initial state (initial position) of the robot was obtained by chaotic preprocessing based on the chaotic model. Chaotic preprocessing improves the diversity of robots' positions and prevents the robots from interfering with each other in their initial state.

In this paper, we use the classic Logistic chaotic mapping as a chaotic model, and a chaotic sequence is:

$$z_{n+1} = \rho z_n (1 - z_n) \quad (2)$$

where ρ represents the Logistic parameter. z is generated after I_c iterations. I_c is Logistic equation iterations. If $z^* = [a_i, b_i]$, then a new chaotic sequence z^* can be obtained by the mapping and de-mapping according to Equations (3) and (4).

$$z_i = (z_i^* - a_i) / (b_i - a_i) \quad (3)$$

$$z_i^* = a_i + z_i(b_i - a_i) \quad (4)$$

The Logistic mapping system is very sensitive to the initial values and Logistic parameter, and different initial values and Logistic parameter result in different chaotic sequences. By studying the trajectory sequence distribution diagram generated by chaotic motion, the appropriate initial value and chaotic parameter can be determined. Figure 4 shows the trajectory sequence distribution diagrams distributed on the interval $[0, 1]$ after 500 Logistic equation iterations for different z_0 and ρ , which is used to find z_0 and ρ for chaotic sequences with good ergodicity and global searching ability.

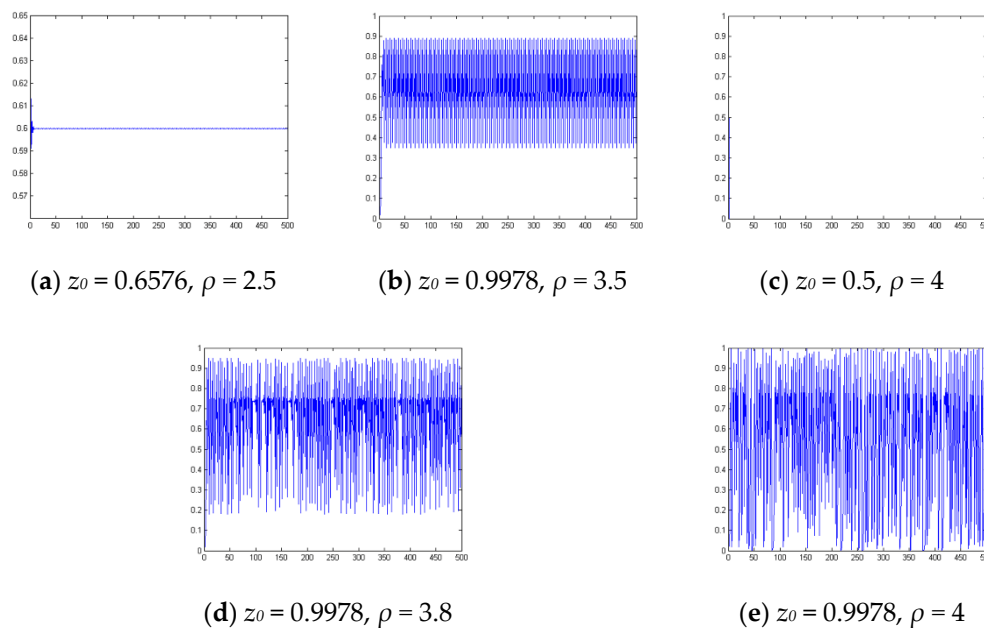


Figure 4. The trajectory sequence distribution diagrams of five combinations of initial values and parameters.

With the initial value $z_0 = 0.9978$ and $\rho = 4$, the Logistic chaotic mapping equation can form an averagely distributed chaotic sequence with good ergodicity, which can search space globally. Therefore, the initial value $z_0 = 0.9978$ and $\rho = 4$ are selected for chaotic model in the following text.

4. Proposed Method

This section describes the proposed coordination strategy of swarm robots for coverage and aggregation inspired by the chemotaxis behavior of *E. coli*.

The main ideas of BCCS are as follows:

- (1) Construct the fitness function, which is available for coverage and aggregation.
- (2) Compare the current fitness function value with previous ones.
- (3) The robot makes running or rotating depending on the comparison result in list (2).

Especially, the coordination control process for aggregation is similar to the reverse coordination control process for coverage. In the following, we present the details of BCCS for coverage and aggregation, respectively.

4.1. The BCCS for Coverage

The robot motion can be modeled as a feedback system, shown in Figure 5. Logistic chaotic mapping is adopted for the chaotic preprocessing module. First, to get more reasonable initial positions of the robots to prevent interference among robots, chaotic preprocessing is needed. Chaotic preprocessing is a process, in which the elements in chaotic sequence z within $[0, 1]$ are converted to that in a new chaotic sequence z^* within $[a, b]$ using Equations (2)–(4). $[a, b]^2$ is the range of horizontal and vertical range of the environment. Moreover, n elements with great differences are extracted from z^* to form a new sequence x' as the initial positions of the robots. $x'_i \in x', x'_i(x'_{i,1}, x'_{i,2})$ is the initial position of robot i . Chaotic preprocessing is optional without considering the interference of the robots.

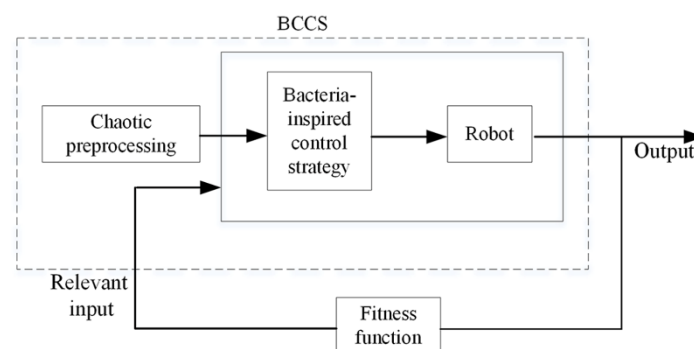


Figure 5. Feedback system architecture for robot motion based bacterial chemotaxis-inspired coordination strategy (BCCS).

The BCCS module is the core of the feedback system. In BCCS, each robot is like a bacterium. BCCS drives robots to generate future actions (running or rotating) and positions. Also, each robot calculates its fitness function. Each robot takes the fitness function value as feedback input.

In this paper, we divided the environment into a large number of cells in the same size. The $Cell_Index = \{1, 2, \dots, l\}$ is the set of cell index. c_k ($k \in Cell_Index$) is the center of the cell k . If $\|c_k - x_i(t)\| < Rs$, then the cell k is covered by robot i at time t , as shown in the left part of Figure 6. Specially, the green cell is covered by robot i , but the yellow cell is not. When the robot i senses the robot j ($d_{ij} = \|x_i - x_j\| < Ri$), overlapping cells are generated, such as the gray cells in the right part of Figure 6. We defined the number of cells only covered by robot i at time t as $F_i(t)$, which also is the fitness function value of robot i , that is:

$$F_i(t) = |O_i(t)| \quad (5)$$

where $|\cdot|$ denotes cardinality of set $O_i(t)$. $O_i(t)$ is the set of cells covered only by robot i at time t , and can be expressed by:

$$O_i(t) = \left\{ k \in Cell_Index \mid \bigcap_{j \in N_i(t)} (\{ \|c_k - x_i(t)\| \leq Rs \} \cap \{ \|c_k - x_j(t)\| > Rs \}) \right\} \quad (6)$$

where $N_i(t)$ is the neighbors of robot i at time t .

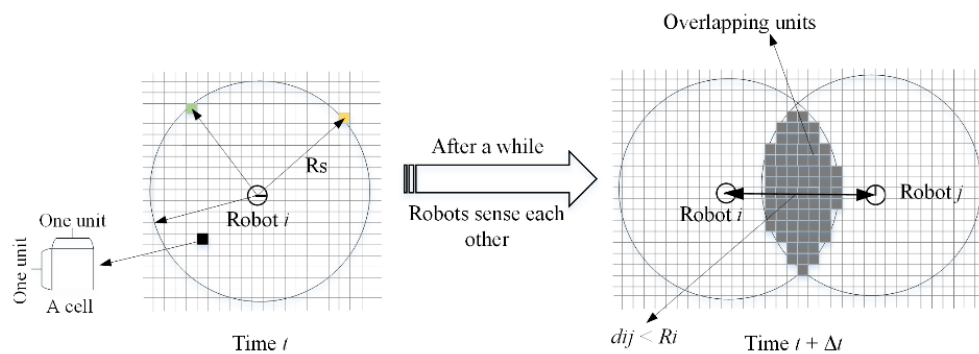


Figure 6. Covered cells and overlapping cells.

Based on the fitness function in Equation (5), we propose the coordinate strategy for coverage:

$$g(x_i(t), [x_j(t)]_{j \in N_i(t)}) \triangleq \begin{cases} \begin{bmatrix} v(t) \\ 0 \end{bmatrix} & \text{if } \zeta_i(t) \leq F_i(t), \\ \begin{bmatrix} v(t) \\ \theta(t) \end{bmatrix} & \text{if } \zeta_i(t) > F_i(t), \end{cases} \quad (7)$$

where $\zeta_i(t) = F_i(t - 1)$ is the fitness function value of the robot i at time $t - 1$. Thus, $\zeta_i(t) \leq F_i(t)$ denotes that the current fitness function value is higher than or equal the previous one. In this case, the robot i maintains the current direction and moves forward. This process is defined as running of robot, which is similar to the swimming of bacterium. In contrast, $\zeta_i(t) > F_i(t)$ denotes that the current fitness function value is lower than the previous one. In this case, the robot i rotates $\theta(t)$ radians, and moves forward. This process is defined as rotating of robot, which is similar to the tumbling of bacterium. $\theta(t)$ is not fixed and is based on the positions of robot i 's neighbors at time t . The rotation direction $\theta(t)$ is calculated using the following equations:

$$\theta(t) = \arccos\left(\frac{\mathbf{r}(t) \cdot \mathbf{v}(t)}{\|\mathbf{r}(t)\| \|\mathbf{v}(t)\|}\right) \quad (8)$$

$$\mathbf{r}(t) = \sum_{j \in N_i} (x_i(t) - x_j(t)) \quad (9)$$

where $\|\cdot\|$ is the norm of vector. $\mathbf{r}$ is the integrated virtual repulsive force acting on the robot i from its neighbors at time t . A simple case is shown in Figure 7. It is evident that the robot moves far away from its neighbors by virtual forces.

A flowchart of BCCS for coverage is shown in Figure 8. Chaotic preprocessing is optional, and A1 is BCCS without chaotic preprocessing. Each robot performs a random walk at its first iteration. After initialization, each robot performs the running or rotating according to the comparison result of fitness function values at the previous moment and the current moment. Especially, the robot can detect the boundary of space E within R_s and then will move forward away from the boundary. Finally, the process will stop when the maximum number of iterations is reached, or the number of covered cells meets the termination conditions. From the above coordinate strategy, each robot tries to maximize its F_i , and the coverage will be achieved.

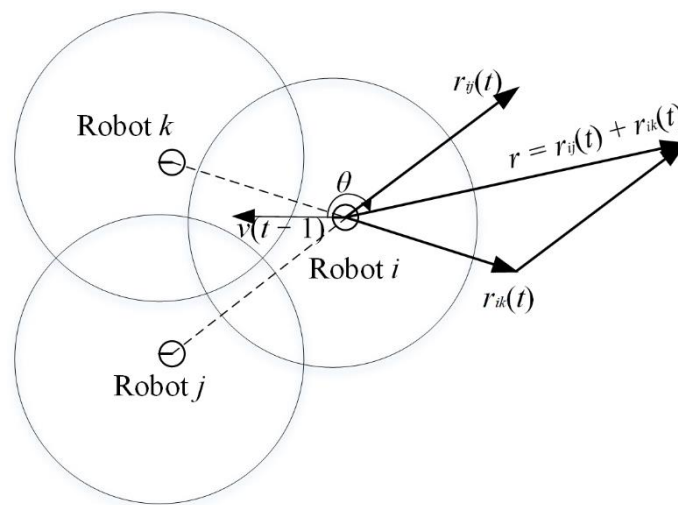


Figure 7. Illustration of the integrated virtual repulsive force for robot i .

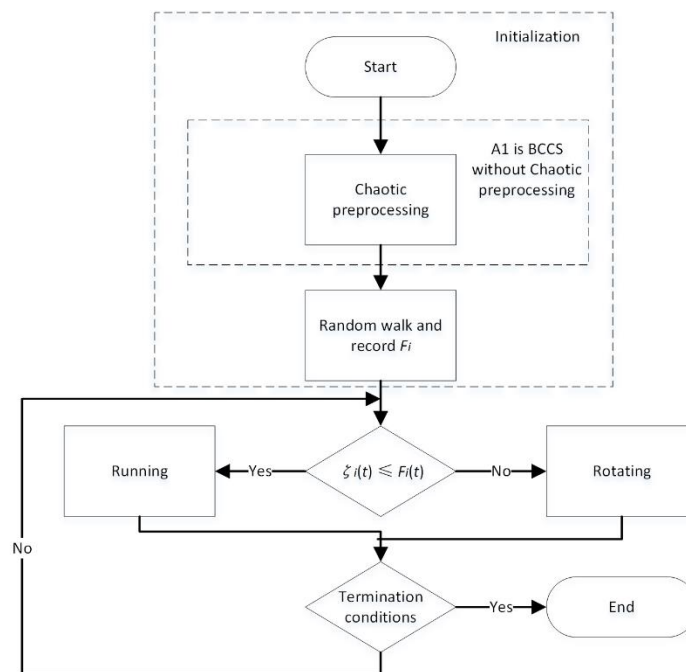


Figure 8. Flowchart of BCCS for coverage.

4.2. The BCCS for Aggregation

The coordinate strategy described in this section is similar to that in Section 4.1, and we introduce a solution to aggregation. The fitness function is obtained by Equations (5) and (6). Two basic behaviors, running and rotating, also exist in the coordinate control strategy for the aggregation. We modify the coordinate strategy for aggregation, as shown in Equation (10).

$$g\left(x_i(t), [x_j(t)]_{j \in N_i(t)}\right) \triangleq \begin{cases} \begin{bmatrix} v(t) \\ 0 \end{bmatrix} & \text{if } \zeta_i(t) \geq F_i(t), \\ \begin{bmatrix} v(t) \\ \theta(t) \end{bmatrix} & \text{if } \zeta_i(t) < F_i(t), \end{cases} \quad (10)$$

where, $\zeta_i(t) \geq F_i(t)$ denotes that the current fitness function value is lower or equal to the previous one. In this case, the robot i maintains the current direction and moves forward.

This process is defined as the running of robot for aggregation. In contrast, $\zeta_i(t) < F_i(t)$ denotes that the current fitness function value is higher than the previous one. In this case, the robot i rotates $\theta(t)$ radians, and moves forward. This process is defined as rotating of robot for aggregation. $\theta(t)$ solution is based on Equations (8), (11) and (12):

$$r(t) = w_1 * \sum_{j \in N_i} (x_j(t) - x_i(t)) + w_2 * rand(0,1) \quad (11)$$

$$w_1 = \frac{Num_{N_i}(t)}{n} \quad (12)$$

where $Num_{N_i}(t)$ is the number of neighbors of robot i at time t . The $rand(0,1)$ is the random factor that follows the (0,1) uniform distribution. The $rand(0,1)$ improve the robot exploration ability to prevent robots from rendezvousing multiple aggregations. The coefficient w_1 and w_2 , $0 \leq w_1, w_2 \leq 1$ and $w_1 + w_2 = 1$, are the weights of the neighbor impact and random impact, respectively. In this coordinate strategy, w_1 and w_2 can be dynamically adjusted according to the number of neighbors of the robot. The larger the value of w_1 (neighbor impact) gets, the smaller w_2 (random impact) is. For example, $w_2 = 0$, means that the robot can sense all the individuals in the swarm after iterations, which causes all the robots to be aggregate quickly.

Figure 9 is flowchart of BCCS for aggregation. Each robot also performances the running or rotating according to the above coordination strategy for aggregation. In aggregation, each robot tries to minimize its F_i and simultaneously monitor the number of neighbors in its sub-aggregation. When robots from multiple aggregations, w_2 can promote the exploration of individuals within each sub-aggregation. Fusion occurs when two sub-aggregations meet, and the aggregation will be achieved. In the whole process, there is no central controller or global network in the swarm robotic system and no direct communication between the robots.

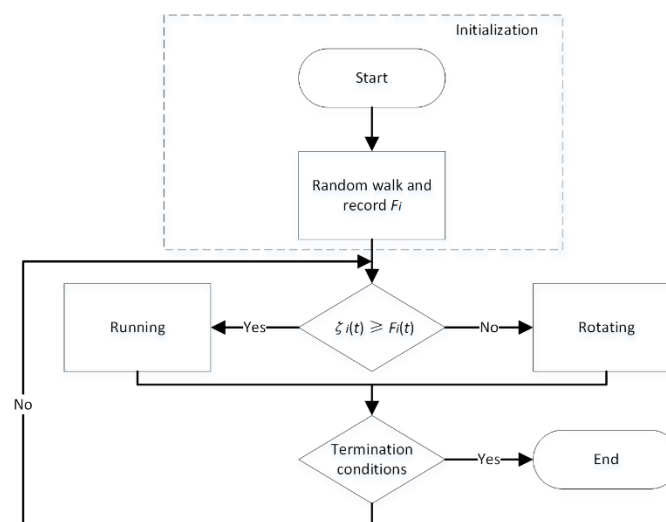


Figure 9. Flowchart of BCCS for aggregation.

5. Simulations and Discussion

This section mainly contains four parts. Section 5.1 presents the simulation setup of parameters. Section 5.2 introduces the evaluation criteria which been used to describe the performances of algorithms. Section 5.3 compares the coverage performance of BCCS with that of the state-of-the-art bacteria-inspired controller in Reference [27]. Section 5.4 compares the aggregation performances of the two algorithms. Each simulation, with respect to different numbers of robots in different sizes of environments, is repeated 30 times for each algorithm because all simulation results are probabilistic.

5.1. Simulations Setup

In this paper, we do not consider the physical structure of the robot, but abstract it as a start point. The environments are unknown and terrain flat, without any external factors. The system parameters are summarized in Table 2.

Table 2. Parameter settings of the swarm robotic system.

Parameter	Value
Environments, E	60 units $\times$ 60 units, 80 units $\times$ 80 units, 100 units $\times$ 100 units
Cell size, k	1 unit $\times$ 1 unit
Number of robots, n	9, 16, 25
Velocity of robot, v	1 unit/iteration
Maximum iterations, I_{max}	2L
Covered radius, R_s	10 units
Sense radius, R_i	20 units
Logistic equation iterations, I_c	500
Rotation range, θ	0–360°

5.2. Evaluation Criteria

Before we discuss the results, let us define some performance measures: (1) F is the sum of F_i for all the robots, which is shown as Equation (13). F corresponds to the performance measure from the perspective of the degree of the covered cells of the individual robot; (2) S is the number of cells covered by all the robots, which is shown as Equation (14). S corresponds to the performance measure from the perspective of the degree of the covered cells of the environment; (3) average and standard deviation of iterations; (4) the number of aggregations is another important performance measure for aggregation.

$$F = \sum_{i \in N} F_i \quad (13)$$

$$S = \left| \left\{ \bigcup_{i \in N} (k \in \text{Cell_Index} \mid \|c_k - x_i\| \leq R_s) \right\} \right| \quad (14)$$

5.3. Comparison for Coverage

In this section, BCCS and A1 are compared with the bacteria-inspired controller in Reference [27], where A1 is the BCCS without chaotic preprocessing. First, we consider an extreme case where all the robots are at the same point (30, 30), and compare A1 with other bacteria-inspired controller. Also, the case of BCCS is studied.

Figure 10 shows the iteration series of the robot positions for A1 and the controller in Reference [27], where the stars in different colors represent different robots. The environment is 60 units $\times$ 60 units. The number of robots is 9, and the radius of the red circle is R_s . The maximum number of iterations is 120. In Figure 10, the robot distribution based on A1 is more uniform than that based on the controller in Reference [27] after 60 iterations. After 80 iterations, the position of each robot based on A1 is fine-tuned. One can see the same scenario in Figure 10, which causes A1 to have a higher performance. Figure 11 shows the evolution of the performance measure F_i of each robot over iterations. We can conclude that the F_i of each robot increases as the number of iterations increases for two methods at the beginning of the coverage. In Figure 11a, the F_i of some robots oscillates over a wide range for the existing bacteria-inspired controllers. As shown in Figure 11b, F_i of each robot for A1 gradually converges after 80 iterations. It is because the robot evaluates the neighbors' positions when performing rotating. In fact, the robot moves far away from its neighbors, and direction is calculated by Equation (8). The smaller the oscillation amplitude for all the robots is, the more stable the swarm robotic system is.

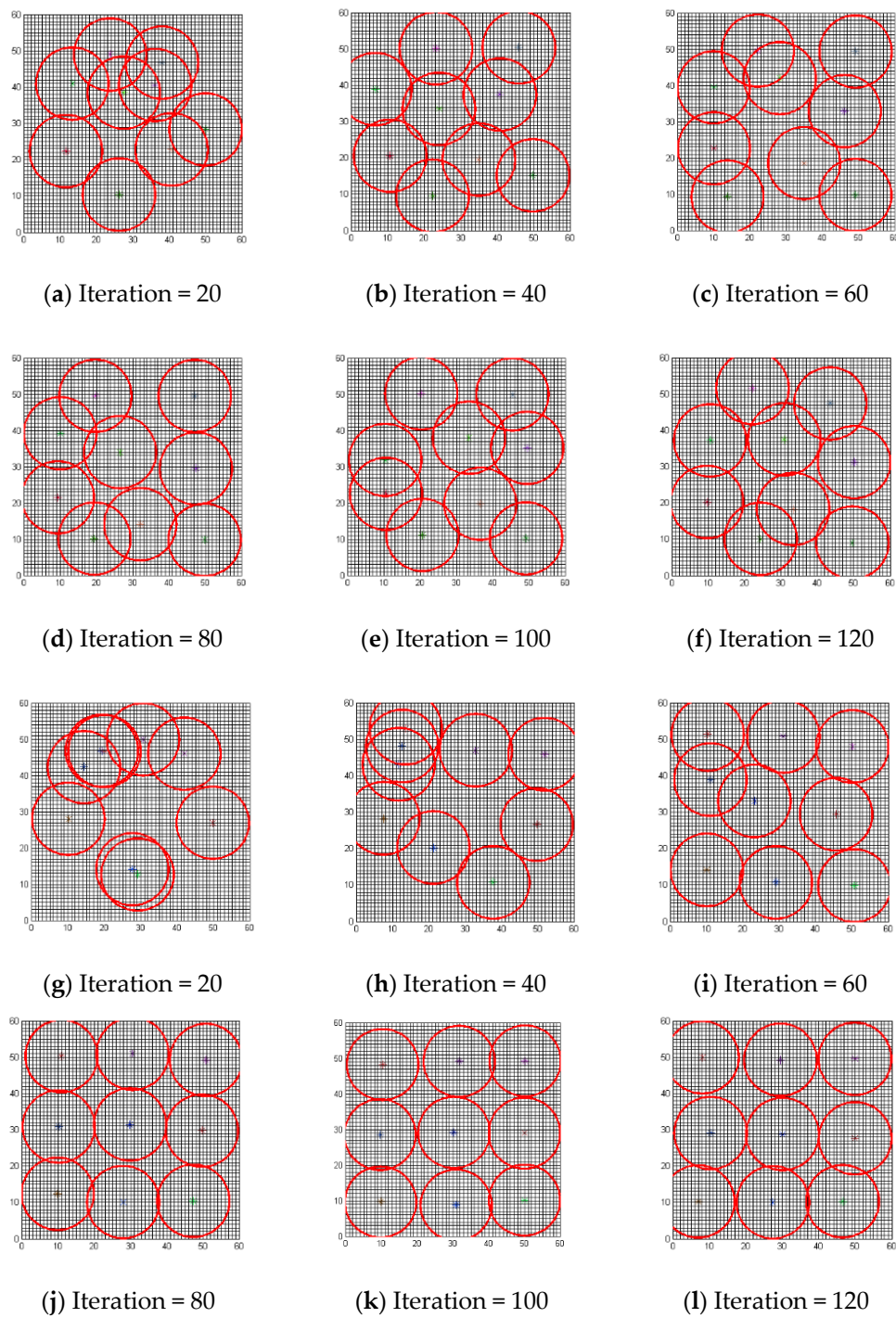


Figure 10. Simulation results for coverage in noise free environment: (a–f) The controller in Reference [27]; (g–l) A1.

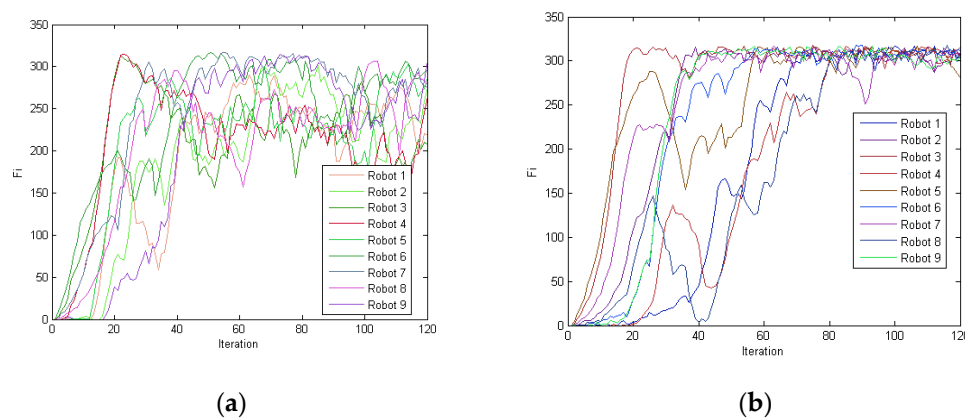


Figure 11. The F_i evolution of each robot for coverage in noise free environment: (a) The controller in Reference [27]; (b) A1.

Next, we consider the noise, and the background noise is fixed at $\sigma_k = 0.005$. $x'(t+1) = x(t+1) + \sigma_k$ is the actual moving position of robot at time $t+1$. Figure 12 shows the evolution of F for all the robots with and without noise. For the noise free environment, although the robot based on the existing bacteria-inspired controller is more exploratory at the first 50 iterations, due to the randomness of the robot's initial motion direction, the robot based A1 can quickly adjust their motion planning as the number of iterations increases. In Figure 12, although the noise reduces the stability of change of F for A1, the performance is better than that for the controller in Reference [27]. Meanwhile, after 60 iterations, the covered area of the environment for A1 is larger than that for the existing bacteria-inspired controller as the number of iterations increases, as shown in Figure 13.

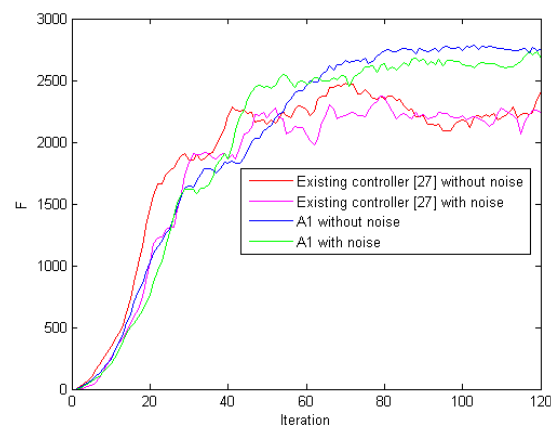


Figure 12. The F evolution of two methods without and with noise for coverage. The red and purple lines are for the controller without and with noise in Reference [27], respectively. The blue and green lines are for A1 without and with noise, respectively.

Next, we also consider chaotic preprocessing, which improves the diversity of positions of robots and the convenience of individual deployment. Some simulations are presented under different numbers of robots in different sizes of noise free environments. Each robot can cover 316 cells. To evaluate iteration consumption, we defined that the condition meets $S/(316n) \geq 95\%$, which means coverage is successfully completed, and record the current number of iterations. The robots' initial positions for the controller in Reference [27], and A1 are randomly generated and the same. However, the robots' initial positions are obtained by chaotic preprocessing in BCCS. The comparison of the average and standard deviation of iterations and the success rate is shown in Table 3. A1 and BCCS have a higher success rate than the existing state-of-the-art bacteria-inspired controller. For example, the success rate of A1 and BCCS for 25 robots is 90.00% and 100%, respectively, while the controller in Reference [27] just has 60%. The BCCS has the lowest number of

iterations. In BCCS, chaotic preprocessing help robots explore quickly in the early stage and fine-tune in the later stage, which shortens the coverage completion time.

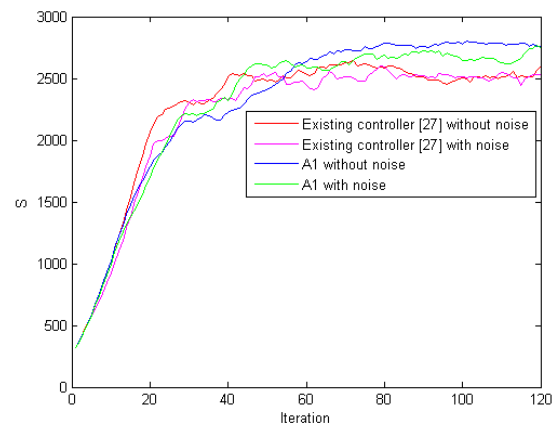


Figure 13. The S evolution of two methods without and with noise for coverage. The red and purple lines are for the controller without and with noise in Reference [27], respectively. The blue and green lines are for A1 without and with noise, respectively.

Table 3. Performance comparison results. Avg = average of iterations. Std = standard deviation of iterations. Succ = success rate.

The Controller in Reference [27]			A1			BCCS			#Robots	#Environment
Avg	Std	Succ (%)	Avg	Std	Succ (%)	Avg	Std	Succ (%)		(Unit $\times$ Unit)
84	10.88	93.33	73	9.35	100	55	8.81	100	9	60 $\times$ 60
117	17.19	76.67	98	13.17	93.33	71	8.47	100	16	80 $\times$ 80
162	15.88	60.00	126	19.55	90.00	108	22.88	100	25	100 $\times$ 100

5.4. Comparison for Aggregation

We employ the BCCS given by Equations (5), (6) and (10)–(12) for aggregation. We assume that the systems for the controller in Reference [27] and BCCS are distributed, and the robots are not connected by a graph G . The simulation configuration is the same as that used for coverage. Figure 14 shows the iteration series of the robot positions for the controller in Reference [27], and BCCS, where robots have dispersed initial positions, as shown in Table 4. In Figure 14, the controller in Reference [27] faces the problem of the slow aggregation process. Although the robots aggregate locally, the fusion of sub-aggregations is a very slow process. After the maximum number of iterations, there are three sub-aggregations in the environment, which means the aggregation failure. However, the robots based on BCCS aggregated in a common place at 60 iterations. From 60 iterations to 120 iterations, the aggregation structure is stable. This is because the BCCS can balance cohesive swarms and exploration of individuals when performing rotating. The w_2 helps a sub-aggregation explore space and search other sub-aggregations. The higher w_2 means the more obvious the exploration capability of the robot in a sub-aggregation. When the robot senses the others with R_i , w_2 is zero, and the robot loses the exploration capabilities and quickly gathers to its neighbors. Figure 15 shows the evolution of the performance measure F_i of each robot over iterations for two methods. In Figure 15b, the F_i of each robot for BCCS gradually converges after 60 iterations. In fact, each robot will not escape from its current aggregation because the w_2 of each robot is zero.

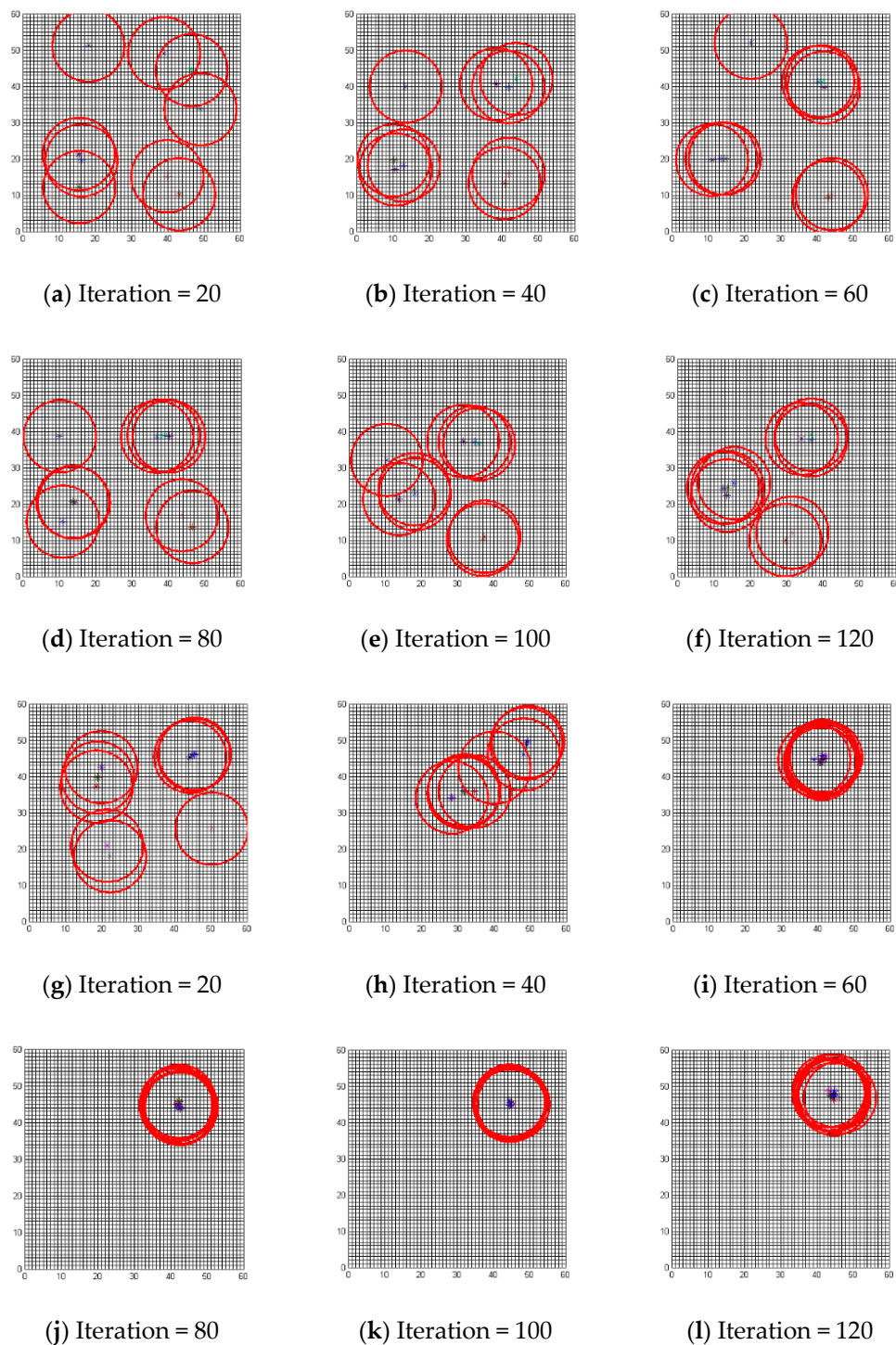


Figure 14. Simulation results for aggregation in noise free environment: (a–f) The controller in Reference [27]; (g–l) BCCS.

Figure 16 shows the F evolution of two methods with and without noise. We find that the F value for BCCS decreases faster than the existing controller in Reference [27], which means the robot can move more quickly toward their neighbors. The noise makes the change of F for the controller in Reference [27] produce obvious fluctuations. However, the noise prompts all robots based on BCCS to aggregate quickly, because the noise improves the robot's exploration ability. When the robots aggregate in a common place, the noise impact is not enough to make robots escape the current aggregation because of the attraction of neighbors. In Figure 17, the covered cells for BCCS with noise are 358 at 53 iteration, which is 1.13 times that for a single robot. The covered cells for BCCS without noise are

354 at 60 iterations. That is to say, BCCS is robust to noise, and all robots rendezvous in a common place.

Table 4. Robots' initial positions.

Robot	$x_{i,1}$	$x_{i,2}$
1	10	10
2	30	10
3	50	10
4	10	30
5	30	30
6	50	30
7	10	50
8	30	50
9	50	50

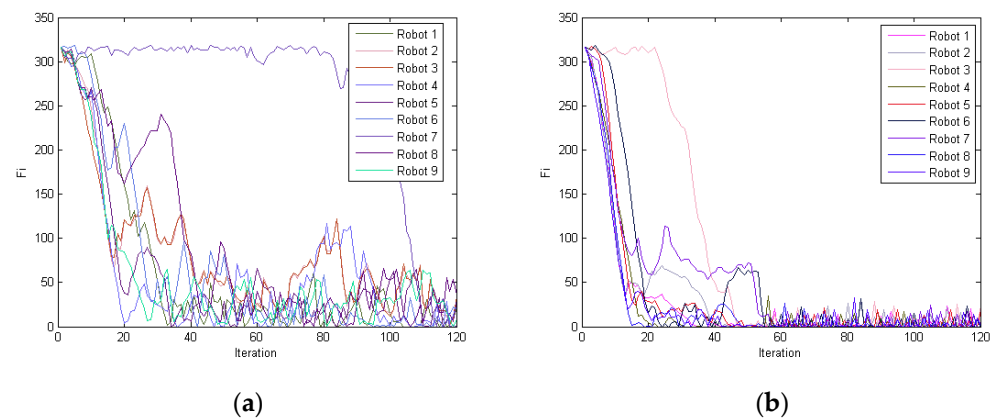


Figure 15. The F_i evolution of each robot for aggregation in noise free environment: (a) The controller in Reference [27]; (b) BCCS.

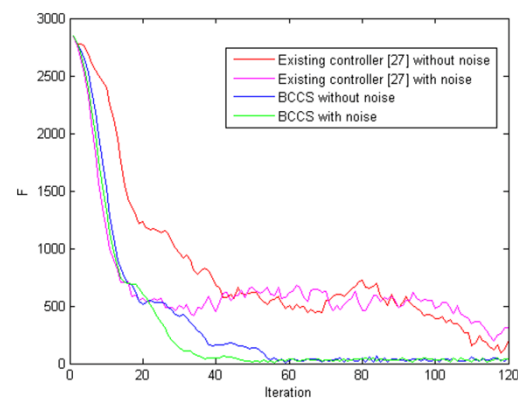


Figure 16. The F evolution of two methods without and with noise for aggregation. The red and purple lines are for the controller without and with noise in Reference [27], respectively. The blue and green lines are for BCCS without and with noise, respectively.

Table 5 shows the simulation results of two methods for aggregation in a noise free environment. It is worth mentioning that the initial positions of robots for the two methods are randomly generated in local space and the same. Although all simulation results are probabilistic for two methods, we find the superiority of BCCS through some simulations. The completion condition of aggregation is $S/316 \leq 1.5$, which means all robots rendezvous in a common place. The simulation results are that BCCS has a significant performance comparing with the existing controller for all schemes. For example, the

number of iterations of BCCS is lower than that of the controller in Reference [27] under different numbers of robots. With the increasing number of robots, the success rate of BCCS for all schemes holds on 100%, in contrast, the success rate of controllers in Reference [27] is 46.67%, 30%, and 23.33%, respectively. It is clear that BCCS has a more significant advantage as the number of robots increases. It also means that BCCS has a certain scalability. Figure 18 is the comparison of the number of aggregations for two methods in detail. The number of aggregations for BCCS always 1, while the number of the controller in Reference [27] is from 1 to 5. This further confirms the applicability and reliability of BCCS.

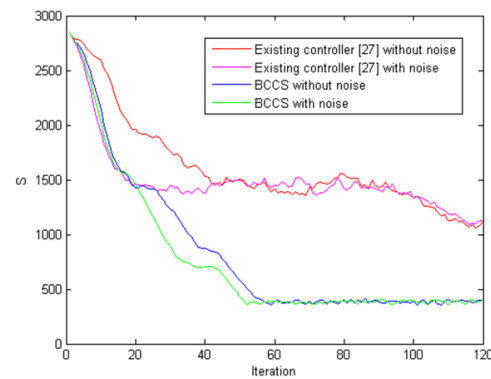


Figure 17. The S evolution of two methods without and with noise for aggregation. The red and purple lines are for the controller without and with noise in Reference [27], respectively. The blue and green lines are for BCCS without and with noise, respectively.

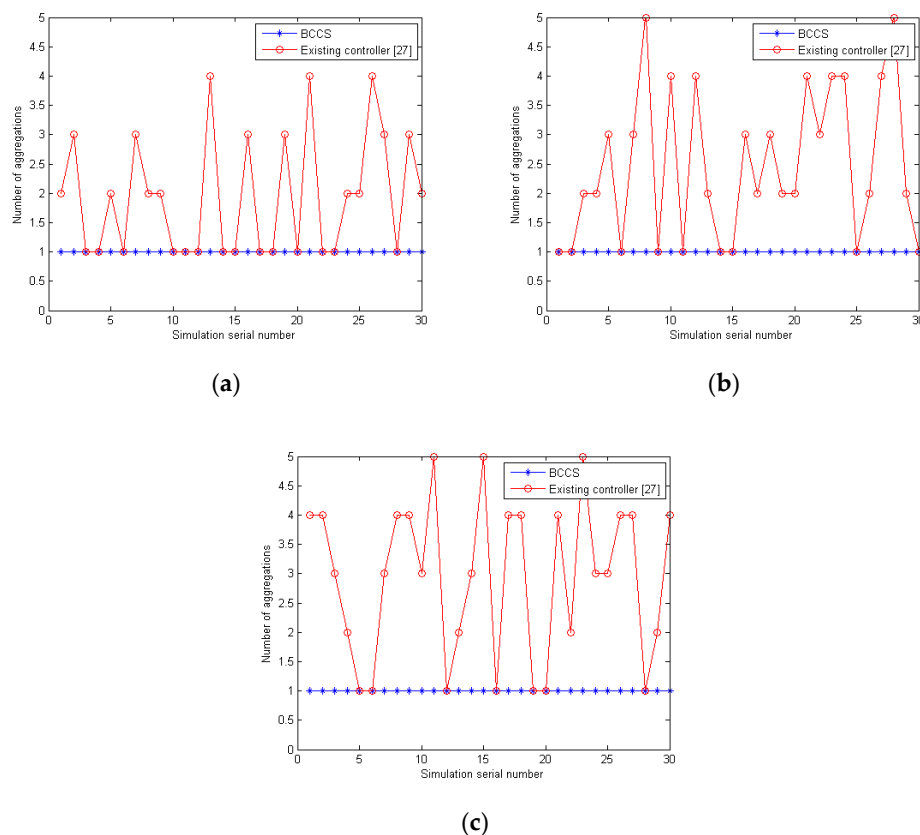


Figure 18. The comparison of the number of aggregations: (a) $n = 9$; (b) $n = 16$; (c) $n = 25$. (a–c) are three simulation schemes under different number of robots. The vertical coordinate values of blue star points for BCCS maintain 1 for thirty simulations. The vertical coordinate values of the red circle for the controller in Reference [27] are from 1 to 5.

Table 5. Performance comparison results.

The Controller in Reference [27]			BCCS			#Robots	#Environment
Avg	Std	Succ (%)	Avg	Std	Succ (%)		(Unit × Unit)
104	8.53	46.67	56	9.85	100	9	60 × 60
137	8.26	30.00	100	14.67	100	16	80 × 80
178	10.36	23.33	125	7.76	100	25	100 × 100

6. Conclusions and Future Work

In this paper, a bacterial chemotaxis-inspired coordination strategy (BCCS) of a swarm robotic system is proposed for coverage and aggregation. BCCS has three advances that yield better performance than that of the existing state-of-the-art bacteria-inspired controller. First, chaotic preprocessing can provide better initial positions of the robots for coverage. Second, BCCS has better exploration capability for coverage. Third, the system is distributed, and the robots do not need to be connected by a global graph for aggregation. The BCCS balances cohesive swarms and exploration of individuals for aggregation, which makes robots rendezvous in a common place. Simulations results show that BCCS has fewer iterations and a higher success rate in most cases. In the future, we will examine the performance change in an irregular environment with obstacles by modifying the model or dividing the environment into some areas without obstacles. Finally, we will focus on applying BCCS to a real swarm robotic system.

Author Contributions: Methodology, L.J. and H.M.; software, L.J.; validation, L.J. and P.T.; writing—original draft preparation, L.J.; writing—review and editing, L.J. and H.M.; supervision, H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National key research and development program of China [grant number 2018AAA0102700].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dadgar, M.; Jafari, S.; Hamzeh, A. A PSO-Based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing* **2016**, *177*, 62–74. [\[CrossRef\]](#)
2. Tang, H.; Sun, W.; Yu, H.; Lin, A.; Xue, M.; Song, Y. A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments. *Appl. Intell.* **2019**, *49*, 2603–2622. [\[CrossRef\]](#)
3. Aguilar, G.; Bravo, L.; Ruiz, U.; Murrieta-Cid, R.; Chavez, E. A Distributed Algorithm for Exploration of Unknown Environments with Multiple Robots. *J. Intell. Robot. Syst.* **2019**, *95*, 1021–1040. [\[CrossRef\]](#)
4. De Rango, F.; Palmieri, N.; Yang, X.; Marano, S. Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. *Soft Comput.* **2018**, *22*, 4251–4266. [\[CrossRef\]](#)
5. Oyekan, J.; Hu, H.; Gu, D. Exploiting bacteria swarms for pollution mapping. In Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Gui Lin, China, 19–23 December 2009; pp. 39–44.
6. Oyekan, J.; Gu, D.B.; Hu, H.S. Visual Imaging of Invisible Hazardous Substances Using Bacterial Inspiration. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 1105–1115. [\[CrossRef\]](#)
7. Oyekan, J.; Gu, D.B.; Hu, H.S. A model for using self-organized agents to visually map environmental profiles. *Ecol. Complex.* **2014**, *19*, 68–79. [\[CrossRef\]](#)
8. Sourjik, V.; Wingreen, N.S. Responding to Chemical Gradients: Bacterial Chemotaxis. *Curr. Opin. Cell Biol.* **2012**, *24*, 262–268. [\[CrossRef\]](#)
9. Bakhale, M.; Hemalatha, V.; Dhanalakshmi, S.; Kumar, R.; Jain, M.S. A Dynamic Inertial Weight Strategy in Micro PSO for Swarm Robots. *Wirel. Pers. Commun.* **2020**, *110*, 573–592. [\[CrossRef\]](#)

10. Martel, S.; Mohammadi, M. Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 500–505.
11. Servant, A.; Qiu, F.; Mazza, M.; Kostarelos, K.; Nelson, B.J. Controlled In Vivo Swimming of a Swarm of Bacteria-Like Microrobotic Flagella. *Adv. Mater.* **2015**, *27*, 2981–2988. [\[CrossRef\]](#)
12. Yigit, B.; Alapan, Y.; Sitti, M. Programmable Collective Behavior in Dynamically Self Assembled Mobile Microrobotic Swarms. *Adv. Sci.* **2019**, *6*, 1801837. [\[CrossRef\]](#)
13. Nurzaman, S.G.; Matsumoto, Y.; Nakamura, Y.; Koizumi, S.; Ishiguro, H. “Yuragi”-Based adaptive mobile robot search with and without gradient sensing: From bacterial chemotaxis to a Levy walk. *Adv. Robot.* **2011**, *25*, 2019–2037. [\[CrossRef\]](#)
14. Nurzaman, S.G.; Matsumoto, Y.; Nakamura, Y.; Shirai, K.; Ishiguro, H. Bacteria-Inspired underactuated mobile robot based on a biological fluctuation. *Adapt. Behav.* **2012**, *20*, 225–236. [\[CrossRef\]](#)
15. Shirai, K.; Matsumoto, Y.; Nakamura, Y.; Koizumi, S.; Ishiguro, H. Noise-Based Underactuated Mobile Robot Inspired by Bacterial Motion Mechanism. In Proceedings of the 2009 IEEE-RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 11–15 October 2009; pp. 1043–1048.
16. Davison, T.; Nguyen, H.; Nickels, K.; Frasn, D.; Basagaoglu, H. Localization of chemical sources using *E. coli* chemotaxis. In Proceedings of the Conference on Bioinspiration, Biomimetics and Bioreplication, Las Vegas, NV, USA, 21–22 May 2016; p. 9797.
17. Russell, R.A.; Bab-Hadiashar, A.; Shepherd, R.L.; Wallace, G.G. A comparison of reactive robot chemotaxis algorithms. *Robot. Auton. Syst.* **2003**, *45*, 83–97. [\[CrossRef\]](#)
18. Nickels, K.; Nguyen, H.; Frasn, D.; Davison, T. Effective Exploration Behavior for Chemical-Sensing Robots. *BIOMIM* **2019**, *4*, 69. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Hossain, M.A.; Ferdous, I. Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. *Robot. Auton. Syst.* **2014**, *64*, 137–141. [\[CrossRef\]](#)
20. Jimenez, F.; Gonzalez-Herrera, F.; Lara-Alvarez, C. A Coordinated Wilderness Search and Rescue Technique Inspired by Bacterial Foraging Behavior. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics, Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 318–324.
21. Sharma, S.; Sur, C.; Shukla, A.; Tiwari, R. Multi Robot Path Planning for Known and Unknown Target Using Bacteria Foraging Algorithm. In Proceedings of the 5th International Conference on Swarm, Evolutionary and Memetic Computing, Bhubaneswar, India, 18–20 December 2015; pp. 674–685.
22. Liang, X.D.; Li, L.Y.; Wu, J.G.; Chen, H.N. Mobile robot path planning based on adaptive bacterial foraging algorithm. *J. Cent. South Univ.* **2013**, *20*, 3391–3400. [\[CrossRef\]](#)
23. Cortes, J.; Martinez, S.; Karatas, T.; Bullo, F. Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* **2004**, *20*, 243–255. [\[CrossRef\]](#)
24. Bayindir, L. A review of swarm robotics tasks. *Neurocomputing* **2016**, *172*, 292–321. [\[CrossRef\]](#)
25. Garnier, S.; Jost, C.; Gautrais, J.; Asadpour, M.; Caprari, G.; Jeanson, R.; Grimal, A.; Theraulaz, G. The embodiment of cockroach aggregation behavior in a group of micro-robots. *Artif. Life* **2008**, *14*, 387–408. [\[CrossRef\]](#)
26. Caratozzolo, M.; Carnazza, S.; Fortuna, L.; Frasca, M.; Guglielmino, S.; Gurrieri, G.; Marletta, G. Self-Organizing models of bacterial aggregation states. *Math. Biosci. Eng.* **2008**, *5*, 75–83.
27. Izumi, S.; Azuma, S.I.; Sugie, T. Multi-Robot Control Inspired by Bacterial Chemotaxis: Coverage and Rendezvous via Networking of Chemotaxis Controllers. *IEEE Access* **2020**, *8*, 124172–124184. [\[CrossRef\]](#)
28. Izumi, S.; Azuma, S.I. Chemotaxis-Inspired Control for Multi-Agent Coordination: Formation Control by Two Types of Chemotaxis Controllers. *New Gener. Comput.* **2020**, *38*, 303–324. [\[CrossRef\]](#)
29. Marques, L.; Nunes, U.; de Almeida, A. Olfaction-Based mobile robot navigation. *Thin Solid Films* **2001**, *418*, 51–58. [\[CrossRef\]](#)
30. Shklarsh, A.; Ariel, G.; Schneidman, E.; Ben-Jacob, E. Smart Swarms of Bacteria-Inspired Agents with Performance Adaptable Interactions. *PLoS Comput. Biol.* **2011**, *7*, e1002177. [\[CrossRef\]](#)
31. Rashid, S.; Long, Z.C.; Singh, S.; Kohram, M.; Vashistha, H.; Navlakha, S.; Salman, H.; Oltvai, Z.N.; Bar-Joseph, Z. Adjustment in tumbling rates improves bacterial chemotaxis on obstacle-laden terrains. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 11770–11775. [\[CrossRef\]](#)
32. Rashid, S.; Singh, S.; Navlakha, S.; Bar-Joseph, Z. A bacterial based distributed gradient descent model for mass scale evacuations. *Swarm Evol. Comput.* **2019**, *46*, 97–103. [\[CrossRef\]](#)
33. Kurdi, H.; AlDaood, M.F.; Al-Megren, S.; Aloboud, E.; Aldawood, A.S.; Youcef-Toumi, K. Adaptive task allocation for multi-UAV systems based on bacteria foraging behavior. *Appl. Soft. Comput.* **2019**, *83*, 105643. [\[CrossRef\]](#)
34. Pang, B.; Song, Y.; Zhang, C.J.; Wang, H.L.; Yang, R.T. A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method. *J. Robot.* **2019**, *2019*, 6914212. [\[CrossRef\]](#)
35. Yang, B.; Ding, Y.S.; Jin, Y.C.; Hao, K.R. Self-Organized swarm robot for target search and trapping inspired by bacterial chemotaxis. *Robot. Auton. Syst.* **2015**, *72*, 83–92. [\[CrossRef\]](#)
36. Yuan, X.F.; Zhang, T.; Xiang, Y.Z.; Dai, X.S. Parallel chaos optimization algorithm with migration and merging operation. *Appl. Soft. Comput.* **2015**, *35*, 591–604. [\[CrossRef\]](#)