symmetry

MDPI

Article

Dynamic Task Allocation Method of Swarm Robots Based on Optimal Mass Transport Theory

Qiuzhen Wang [1,2] and Xinjun Mao [1,*]

[1] College of Computer, National University of Defense Technology, Changsha 10073, China; qzwang@xtu.edu.cn
[2] School of Computer Science & School of Cyberspace Science, Xiangtan University, Xiangtan 411105, China
[*] Correspondence: xjmao@nudt.edu.cn

Received: 24 September 2020; Accepted: 9 October 2020; Published: 14 October 2020

Abstract: It is difficult for swarm robots to allocate tasks efficiently by self-organization in a dynamic unknown environment. The computational cost of swarm robots will be significantly increased for large-scale tasks, and the unbalanced task allocation of robots will also lead to a decrease in system efficiency. To address these issues, we propose a dynamic task allocation method of swarm robots based on optimal mass transport theory. The problem of large-scale tasks is solved by grouping swarm robots to complete regional tasks. The task reallocation mechanism realizes the balanced task allocation of individual robots. This paper solves the symmetric assignment between robot and task and between the robot groups and the regional tasks. Our simulation and experimental results demonstrate that the proposed method can make the swarm robots self-organize to allocate large-scale dynamic tasks effectively. The tasks can also be balanced allocated to each robot in the swarm of robots.

Keywords: swarm robots; optimal mass transport (OMT) theory; dynamic task allocation; task reallocation; balanced allocation

1. Introduction

Swarm robotics has been defined as "the study of how large numbers of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment" [1]. The task allocation of swarm robots mainly studies how to use a system composed of multiple robots to complete a series of tasks [2–4]. The tasks either cannot be achieved by one robot or can be done more effectively by many robots. Examples of practical applications for swarm robotic systems include cooperative transport [5]; collaborative search and rescue [6]; firefighting [7]; battlefield surveillance [8] and so on. Robots must respond to the dynamic tasks on time by self-organization for dynamic task allocation. Furthermore, the robot can autonomously adjust the state and reallocation tasks to ensure that the environment's dynamic changes and tasks will not reduce task completion efficiency or lead to inconsistency [9]. Therefore, swarm robots' task allocation mechanism needs to continually make reasonable adjustments to task allocation in response to the environment's dynamic changes. In a dynamic environment, task allocation of the swarm robotic system has the following main challenges:

- In the dynamic task environment, the pre-determined task allocation strategy will limit the optimization of resources and reduce the task's efficiency to complete, so it is challenging to make adaptive adjustment of task allocation on time in the unknown task environment.
- When the number of dynamic task points is enormous, the cost of computing will increase dramatically.

- In the task allocation, the unbalanced individual robot executing tasks will reduce the utilization rate of system resources and the efficiency of task execution.

Given the above challenges, this paper proposes a task allocation method based on the optimal mass transport theory in the dynamic environment. The optimal allocation between swarm robots and detected tasks is realized by solving the optimal mass transport formula. The main contributions of this paper are as follows:

1. The Optimal Mass Transport formulation is tailored for the dynamic task allocation of the swarm robots in the unknown environment.
2. In this paper, many task points can be clustered into a series of regional tasks dynamically, and the swarm robots can also be grouped adaptively according to the number of task points in the regional task. Then the method can dramatically reduce the enormous amount of computation when the number of tasks is significant.
3. This paper also proposes a task reallocation mechanism based on the task pool, which realizes the balanced task allocation of individual robots and improves the robots' utilization rate.

The paper is organized as follows. Section 2 gives the related works, and Section 3 describes the problem. Section 4 describes the task allocation mechanism of swarm robots based on optimal mass transport (OMT), and Section 5 is the regional task allocation mechanism based on grouping OMT. Section 6 presents the experiment and analysis, and Section 7 outlines the research conclusions and future works.

2. Related Works

The task allocation mechanism of swarm robots is mainly to improve the system's task execution efficiency and resource utilization, which can be regarded as an optimization allocation problem. Shima et al. [10] solved the multi-task and multi-target problem of UAV based on a genetic algorithm, which can have a good effect on this NP-hard problem by comprehensively considering the specific requirements under unique scenes. Zheng et al. [11] proposed a multi-robot task allocation method based on the ant colony algorithm. However, this method must work in global coordinates, and the execution efficiency will be significantly reduced when the system size increases. The dynamic tabu search method [12], the graph matching algorithm [13], and the distributed auction algorithm [14] can also be used for the task allocation of swarm robots. However, they are only applicable to static environments with unchanged targets.

A dynamic task allocation method was proposed for a heterogeneous robot system based on the parallel auction algorithm, which solved the multi-type task allocation problem in the medical environment [15]. However, this system's computing cost is too expensive, and the coupling degree between task constraints is too high. Lu Wang et al. [16] proposed an improved genetic algorithm for task assignment in a multi-agent environment, which greatly improved the system's execution efficiency and reliability. However, the algorithm was too sensitive to the initial conditions and the number of ants and was not easy to converge. According to the dynamic task environment, Zhu et al. [17] put forward a task allocation method based on the self-organization neural network. By fixing the number of nodes in the neural network's output layer, modifying neighborhood functions, adjusting weights, and updating rules, the method combined the motion planning of the swarm robot system with task assignment. This method achieves high task completion efficiency and robustness. Nevertheless, it is needed to know the global position coordinate of the robot. A group decision mechanism was proposed for task selection under the robot's new task execution time, which significantly improved the system's task completion efficiency [18]. Mendona et al. [19] proposed a simple and effective distributed control algorithm to realize swarm robots' dynamic task allocation. The algorithm is based on decentralized management to achieve and maintain some predefined task allocation ratio. The decision-making process is executed independently by each robot, and the process is simple and straightforward. However, the total number of tasks in these two algorithm experiments is small, which cannot adapt to the large-scale task. Brutschy [20] designed a self-organizing task allocation method that can well realize the optimal allocation of coupled tasks. However, this study is only limited to the scenario with a small number of tasks, and the balance of task allocation for swarm robots is not considered. Ghassemi et al. [21] developed a novel computationally efficient decentralized algorithm, which solved the centralized approach's scalability and the vulnerability to communication disruptions. This method is also applicable to large-scale tasks, while the task allocation balance was not considered. Garapati et al. [22] proposed a task allocation strategy for swarm robots based on game theory, which achieved optimal task assignment and solved the conflict problem of swarm robots. However, this method needs to know the location of all robots and task points in advance, and it cannot realize the balanced assignment of tasks.

Based on the analysis of existing works, it is concluded that there are some limitations in the dynamic task allocation of swarm robots: (1) The swarm robots cannot efficiently self-organize dynamic task allocation in the unknown environment. (2) The swarm robots cannot be assigned tasks on a large scale. (3) The balanced task allocation of the individual robot in the swarm robots is not considered.

3. Problem Description

In the dynamic task environment, the swarm robots continuously detect unknown dynamic task points through limited perception. As shown in Figure 1, both the tasks and the robots are assumed to be a point in a two-dimensional plane. The swarm robotic system will allocate the tasks that were detected. When a new task is detected, the swarm robots need to adjust the task allocation strategy in a timely manner. The scale of the swarm robotic system, the number of tasks, the spatial distribution of tasks, and other factors will affect the task completion efficiency and resource utilization rate of swarm robots.

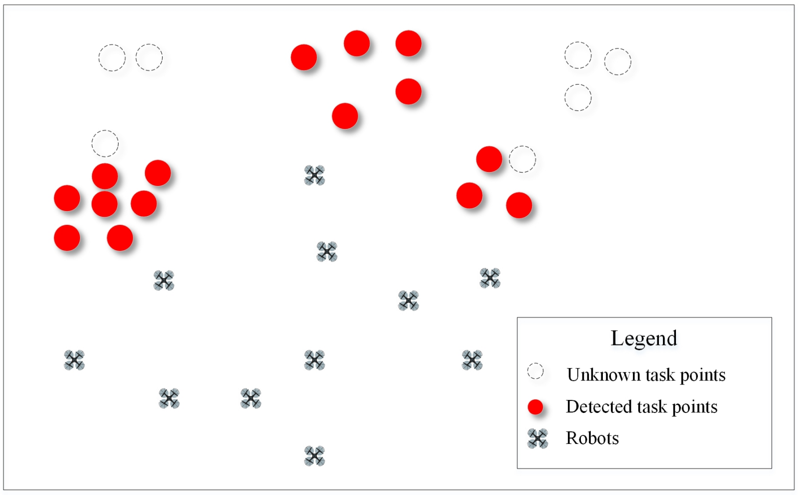


Figure 1. Swarm robot system to perform tasks in the dynamic environment. The solid red circle is the detected task point, and the dashed hollow circle is the unknown task point to be detected. The swarm robots detect the task in the effective perception range and allocate the detected tasks.

From the robotics perspective, "global coordinates" means that there is the origin of a fixed coordinate in the robot's working environment. Each robot has fixed coordinates for positioning, making it easy to acquire the robot's position information. However, it is usually appropriate for a swarm robot system to use "local coordinates" because of its local perception and local communication ability. There is no fixed coordinates origin in the environment, and each robot can only use its origin as a reference to acquire the relative position information of other robots.

To explain the dynamic task allocation problem of swarm robotic system in this paper, the following definitions are needed.

Definition 1. *(Environment) The environment is defined as a set of points E, where* $E \subseteq \mathbb{R}^2$ *and* $\mathbb{R}^2$ *is 2D continuous space.*

Definition 2. *(Swarm robotic system) There are n homogeneous robots in the environment E defined as* $A = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$.

Definition 3. *(Task) There are m tasks in the environment defined as* $\mathbb{T} = \{\mathbb{t}_1, \mathbb{t}_2, \cdots, \mathbb{t}_m\}$. *Each task* $\mathbb{t}_i$ *is an independent and fixed point in a 2D Euclidean space. This paper takes no account of the correlation between tasks, and the task type is the single-robot task that a robot can complete when it reaches the task point.*

Definition 4. *(Task completion time) Task completion time T refers to the time from the time when the robot begins to perform the task until all tasks are completed. If the completion time of task* $\mathbb{t}_i$ *is* t_i*, the completion time of all m tasks is* $\mathrm{T} = \sum_{i=1}^{m} t_i$.

Definition 5. *(Task completion rate) The task* $\mathbb{t}_i$ *is marked as* $\mathbb{t}_i^c$ *when it completes. The task completion rate refers to the ratio of the number of completed tasks* $\left|\sum \mathbb{t}_i^c\right|$ *to the total number of tasks* $|\mathbb{T}|$*, which is defined as* $\eta = \frac{|\sum \mathbb{t}_i^c|}{|\mathbb{T}|} * 100\%$.

Definition 6. *(Total movement distance of the swarm robots) The movement distance of the robot reflects the energy consumption cost of the robot. If the total movement distance of the individual robot* α_i *is* d_i*, then the swarm robot's total movement distance with n robots is defined as* $D = \sum_{i=1}^{n} d_i$.

Definition 7. *(Task pool) The task pool is a container for the robot group to complete tasks. It can be defined as a queue* $TP = (\mathbb{t}_1, \mathbb{t}_2, \cdots,)$. *Each individual robot* α *has a task pool* TP_α.

Problem 1. *With a given number of group robots and a given number of dynamic tasks, identify a method that enables the robot to complete dynamic tasks as soon as possible in a self-organizing way.*

The robot's distance to the task point is one of the objects we need to optimize under the condition that all the robots keep constant speed all the time. If a robot maps to only one task in distributing the requirements, it can simply use the optimal path length from the robot to the task as its cost. If a robot maps to multiple tasks, the traveling salesman problem (TSP) path between the robot and those tasks is the cost.

For n robots and m tasks, the optimization objective is determined according to Equation (1).

$$\min_{M} \sum_{i=1}^{n} \sum_{j=1}^{m} d\left(\alpha_i, \mathbb{t}_j\right) \tag{1}$$

where $d\left(\alpha_i, \mathbb{t}_j\right)$ represents the distance between the robot and the task.

Problem 2. *In the context of a vast number of tasks, identify a method that enables reducing the computational complexity of large-scale tasks and improving the completion speed of tasks.*

In the optimization process of task allocation, all robots' distance and all tasks need to be calculated. When the number of tasks is large enough, the calculation cost must be enormous.

Problem 3. *Identify a method that enables the task balance allocation of the individual robot in the swarm robot system.*

Since all robots are homogeneous, an unbalanced allocation of tasks will reduce the swarm robot system's overall work efficiency. Therefore, it is also one of the optimization objects to realize the individual robot's balanced task allocation.

4. Task Allocation Mechanism of Swarm Robots Based on OMT

The Optimal Mass Transport (OMT) theory comes from economics [23]. In short, how can multiple suppliers reasonably provide products to multiple demanders in order to satisfy the demanders while minimizing the cost? As an optimization problem, the goal is to find the mapping between two distributions (or sets) so that the mapping is the least costly for a given metric. The problem of optimal task allocation with capacity constraints has been widely studied in engineering design and resource allocation economics [24]. It is also used to solve the problem of optimal coverage of sensor networks and robots [25–27]. Dong et al. [28] takes scanning and reconstruction of the large-scale indoor scene as the target, and proposes an algorithm based on optimal mass transport theory for multi-robot collaborative exploration and reconstruction of unknown indoor scene model.

The optimal mass transport theory is used to carry out dynamic task allocation of swarm robots, which generally takes robots as suppliers and tasks as demanders. By solving the optimal mass transport equation, the mapping between robots and the detected tasks can be obtained, so that the task completion cost can be minimized.

Specifically, the spatial distribution of robots in the environment is set as the supplier, as shown in Equation (2).

$$\mu_{\text{source}} = \sum_{i=1}^{n} \delta\left(\alpha_i\right), \tag{2}$$

where $\delta\left(\cdot\right)$ is the Dirac function. The source distribution describes the spatial position of the robots.

The spatial distribution of tasks is set as the demander, as shown in Equation (3). The target distribution describes the spatial position of the tasks.

$$\mu_{\text{target}} = \sum_{j=1}^{m} \delta\left(\mathbb{t}_j\right). \tag{3}$$

The optimal planning problem for allocating tasks to robots can be expressed as finding a mapping M that minimizes the function in Equation (4).

$$\arg\min_{M} \int_{x\in\mathbb{R}^2} \gamma(x, M(x))\, d\mu_{\text{source}}\,, \tag{4}$$

where $\gamma(x, M(x))$ is used to estimate the cost of mapping x to $M(x)$ and $M(x) \sim \mu_{\text{target}}$ is the image of $x \sim \mu_{\text{source}}$. The optimal mass transport objective is the minimum of the cost of M.

As shown in Figure 2, if the number of tasks is greater than the number of robots, a robot should map to multiple tasks, the TSP (traveling salesman problem) distance between the robot and those tasks is the distance cost. Considering the equilibrium of task allocation, the number of tasks each robot map is approximately equal. The swarm robots calculate the total minimum distance cost for the detected tasks to carry out the optimal task allocation.

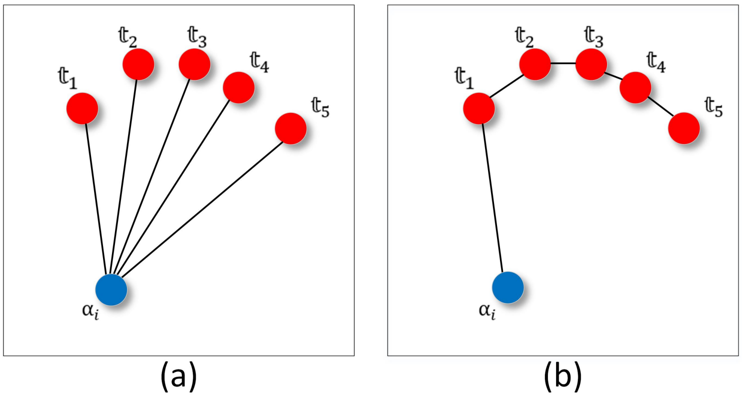


Figure 2. Definition of distance cost based on optimal mass transport (OMT). The red dots are tasks assigned to the robot α_i (bule dot). The lines indicate distance cost. In (**a**), the robot's distance cost is the sum of robot-to-task distances. In (**b**), the robot's distance cost is the traveling salesman problem (TSP)-based distance of the robot to all tasks.

5. Regional Task Allocation Mechanism Based on Grouping OMT

For a large number of tasks and robots, each robot's distance to each task needs to be calculated, and the computational time complexity is $O(n * m)$. The swarm robots cluster tasks into regional tasks and are divided into groups according to the regional tasks, which significantly reduces the cost of task allocation.

Definition 8. *(Regional task)* The regional task Ω is formed by clustering some tasks close to each other. $\Omega = \{\mathbb{t}_1, \mathbb{t}_2, \cdots, \mathbb{t}_k\}$ $(1 \leq k \leq m)$.

If $k \geq 2$, for any task $\mathbb{t}_i$ in the regional task, there is at least one task $\mathbb{t}_\mathrm{j} \in \Omega(i \neq j)$, and the distance between them satisfies the Equation (5).

$$d\left(\mathbb{t}_i, \mathbb{t}_j\right) \leq \theta(i \neq j), \tag{5}$$

where θ is the distance threshold between two tasks. There are two special cases for the regional tasks. If $k = m$, there is only one regional task that all subtasks are in one region. If $k = 1$, the regional task has only one subtask. While the robot detects a new task, the task will be added to the robot's regional task.

Figure 3 shows the mechanism of regional task generation. Initially, swarm robots detect new tasks as they move. Then, the three tasks that were first detected by the different robots became the regional task initiators. Finally, the distance between adjacent tasks is calculated, and the tasks that meet the Equation (5) will be gradually added to the regional task, finally forming three regional tasks.

Generally speaking, only when the number of tasks is large enough is the automatic generation of regional tasks required. In this paper, as long as the number of detected tasks exceeds the number of swarm robots, regional tasks are divided. The primary process of regional task allocation mechanism based on grouping OMT is as follows: (1) the swarm robots divide the detected tasks into several regional tasks. (2) The swarm robots are self-organized and grouped according to the regional tasks. (3) The regional tasks are allocated to the robot groups based on OMT.

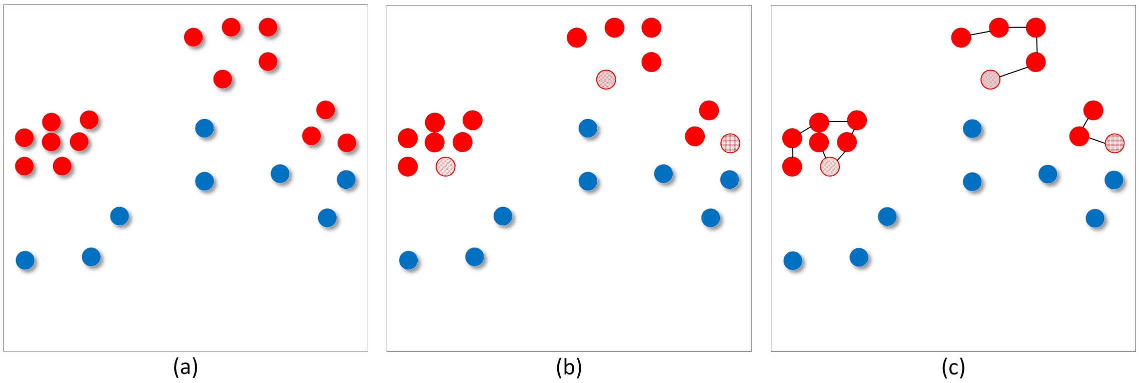


Figure 3. Regional task generation mechanism. (**a**) Swarm robots (the blue dots) detect some new tasks (the red dots). (**b**) Task points first detected by different robots become the regional task's initiators (the dots filled with different patterns). (**c**) Other tasks closer to each other are frequently added to the regional tasks.

5.1. Grouping Task Allocation for Swarm Robots Based OMT

The swarm robots will constantly detect new tasks and expand into regional tasks in the dynamic task environment, so it is necessary to divide the swarm robots into some groups according to the number of regional tasks. Therefore, an improved allocation scheme can be proposed. The robot group has replaced the individual robot as the supplier, and the demander changes from a single task to the regional task. The number of regional tasks is equal to the number of robot groups. To realize the balance of task allocation for the robot, the number of robots in the robot group and the number of tasks in the regional task need to keep the same proportion.

When the number of detected tasks exceeds the swarm robots' number, the swarm robots will divide the tasks into several regional tasks according to the distance between tasks.

Figure 4 shows the self-organizing grouping mechanism of swarm robots. The number of regional tasks determines the number of groups of swarm robots. A robot closest to the regional task center is selected as the leader of the robot group's establishment at first. Then each robot continuously invites the neighbor robots that have not joined any group to join the group until all the robots are in one group by Breadth-First-Search (BFS) algorithm [29]. After the initial grouping of swarm robots is completed, the robot group must be assigned the optimal regional task.

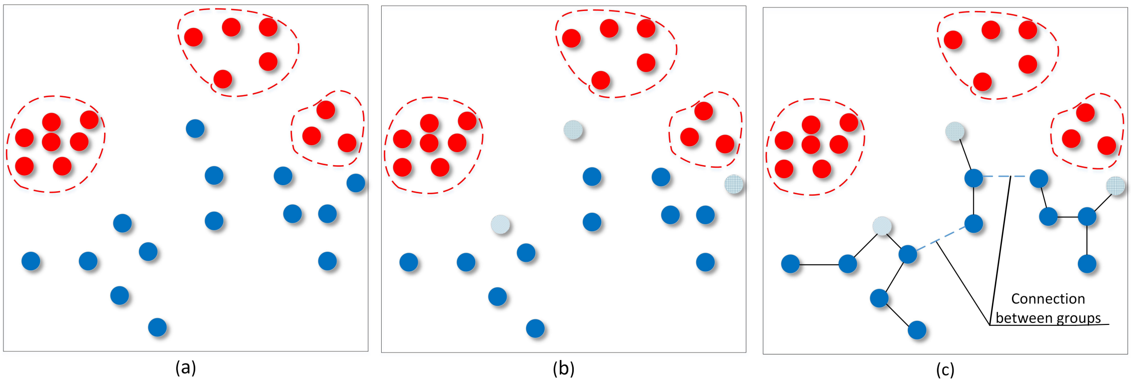


Figure 4. Self-organizing grouping mechanism of swarm robots. The red dots represent detected tasks, and the blue dots represent group robots. (**a**) Initially, regional tasks have been generated. (**b**) The robots closest to each region's tasks become the leaders (the dots filled with a different pattern). (**c**) Grouping starts with the leader robots, the swarm robots self-organize to form several robot groups equal to the number of regional tasks. The groups communicate with each other through the two closest robots in the two groups.

Figure 5 shows the optimal mass allocation of regional tasks based on grouping. The movement distance of the robot is the cost of task allocation through the optimal mass transport theory, and the center point in the regional task is taken as the target point to establish the optimal mass transport equation from the center of robot groups to the regional tasks, as shown in Equation (6).

$$\min_{\mathrm{M}} \sum_{k=1}^{K} d\left(\omega\left(g_k\right), \omega\left(\Omega_k\right)\right), \tag{6}$$

where $\omega(\Omega_k)$ is the center point of the regional task Ω_k and $\omega(g_k)$ is the center point of robot group g_k.

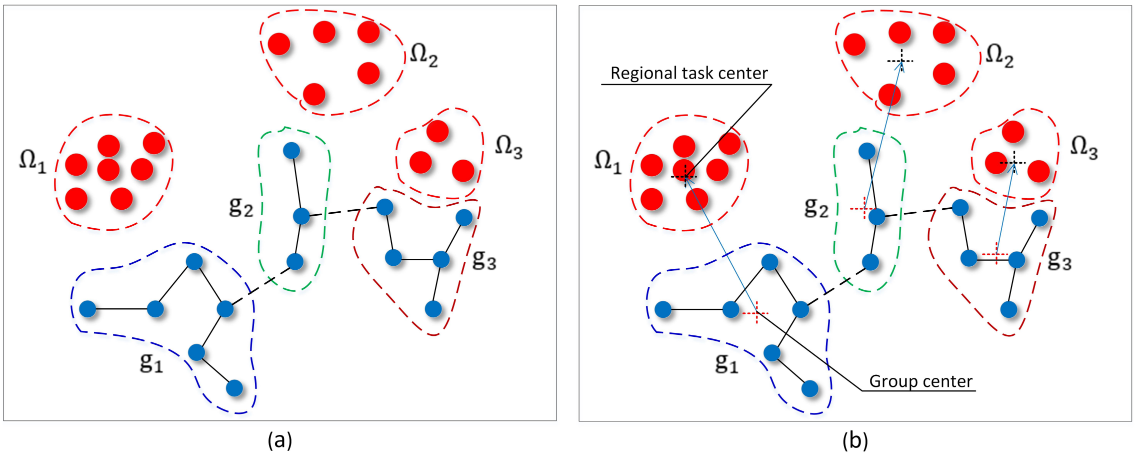


Figure 5. Optimal mass allocation of regional tasks based on grouping. (**a**) Regional tasks and robot groups have been formed. (**b**) Regional tasks are assigned to the robot groups based on OMT.

5.2. *Adaptive Grouping Mechanism of the Swarm Robots*

The imbalance in task allocation will lead to the waste of resources and reduce task completion efficiency. To increase the task completion rate and the resource utilization rate, we propose an adaptive grouping mechanism that can adjust the size of robot groups according to the regional task adaptively. The optimal mass transport optimization objective can be updated to the Equation (7).

$$\min_{M}(\sum_{k=1}^{K} d(\omega(g_k), \omega(\Omega_k)) + \sum_{k=1}^{K}(|\Omega_k|/|g_k| - \sum_{k=1}^{K}|\Omega_k|/\sum_{k=1}^{K}|g_k|)^2). \tag{7}$$

The second term in Equation (7) tries to match the number of regional tasks assigned to the robot groups. Where $|\Omega_k| \,/\, |g_k|$ is the average value of the regional task Ω_k allocated to the individual robot of the robot group g_k, and $\sum_{k=1}^{K}|\Omega_k| \,/\, \sum_{k=1}^{K}|g_k|$ is the average value of all tasks allocated to the individual robot of the swarm robotic system. While $|\Omega_k| \,/\, |g_k|$ is closest to $\sum_{k=1}^{K}|\Omega_k| \,/\, \sum_{k=1}^{K}|g_k|$, the balanced allocation between the robot groups and the regional tasks can be achieved.

The adaptive grouping mechanism of swarm robots is shown in Figure 6. After the swarm robots have completed the regional task allocation, robots will adjust adaptively between the robot groups with an unbalanced size for the regional tasks. As shown in Figure 6a, both robot group g_2 and robot group g_3 are in the state of unbalanced regional task allocation. By minimizing the distance cost between the robot and the center of the regional task, some robots will join another group to achieve the swarm robot system's equilibrium. As shown in Figure 6b, some of the robot group g_3 have joined the robot group g_2.

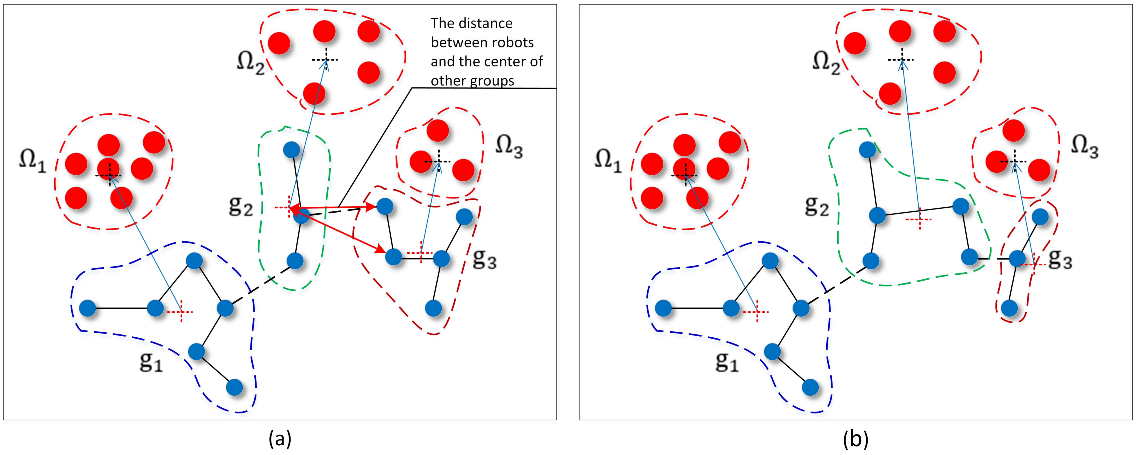


Figure 6. Adaptive grouping adjustment of swarm robots. (**a**) When the regional tasks have been assigned to the robot group, the average number of tasks assigned to the individual robot in the robot group may be larger or smaller than the average of the swarm robot system. The larger robot group will adjust one robot to the smaller robot group (the nearest one). (**b**) After constant adaptive adjustment, each robot group's average number of tasks reaches the system equilibrium.

The adaptive grouping adjustment algorithm is shown in Algorithm 1. The average number of tasks allocated to a robot (line 1) is calculated based on the total number of tasks $\left(\sum_{k=1}^{K}|\Omega_k|\right)$ and the total number of swarm robots $\left(\sum_{k=1}^{K}|g_k|\right)$. All the robot groups without the average task allocation will be found (line 2), and the robot groups g_i that were allocated more tasks than the average will be found (line 3). The robot group g_t which was allocated fewer tasks than the average number and closest to g_i will be found (lines 4–6). The robot in g_i which is nearest to g_t will join g_t until all robot groups meet the average allocation criteria (lines 7–13).

Each robot has roughly the same probability of detecting a new task. When the robot detects a new task, which will be added to the robot's corresponding regional task, and the robot groups will not make adjustments to save the computing costs.

Algorithm 1 Adaptive grouping adjustment algorithm.

```
Input:
    All robot groups $\{g_k\}$ and all detected regional tasks $\{\Omega_k\}$
Output:
    The robot groups $\{g_k\}$ that have been adjusted adaptively
$\lambda = \left\lfloor \sum_{k=1}^{K} |\Omega_k| \,/\, \sum_{k=1}^{K} |g_k| \right\rfloor$;
while $\left(\frac{|\Omega_k|}{|g_k|}\right) < \lambda \vee \left(\frac{|\Omega_k|}{|g_k|}\right) \geq \lambda + 1$ do
    while $g_i \in \{g_k\} \wedge \Omega_i \in \{\Omega_k\} \wedge \left(\frac{|\Omega_i|}{|g_i|}\right) \geq \lambda + 1$ do
        while $g_j \in \{g_k\} \wedge \Omega_j \in \{\Omega_k\} \wedge (j \neq i) \wedge \left(\frac{|\Omega_j|}{|g_j|}\right) < \lambda$ do
            $t \leftarrow \arg\min_{j} \left\| \omega(g_i), \omega(g_j) \right\|$;
        end while
        for $\forall \alpha \in g_i$ do
            $\alpha' \leftarrow \arg\min_{\alpha} \|\alpha, \omega(g_t)\|$;
            $g_t \leftarrow g_t \cup \{\alpha'\}$;
            $g_i \leftarrow g_i - \{\alpha'\}$;
            if $\left(\left(\frac{|\Omega_i|}{|g_i|} \in [\lambda, \lambda + 1]\right) \vee \left(\frac{|\Omega_t|}{|g_t|} \in [\lambda, \lambda + 1]\right)\right)$ then
                break;
            end if
        end for
    end while
end while
```

5.3. Balanced Reallocation of Tasks

Although the scale balance between the robot group and the regional task has been achieved, the individual robot's task in the robot group may not be able to allocate the balance. Therefore, it is necessary to reallocate the task to the robots in the robot group so that each robot can be allocated to the same number of tasks. Since the distance between tasks in the region is not long, it is challenging to distinguish tasks by distance, and a mechanism based on the task pool is proposed to realize the task reallocation of the robot.

Due to the uncertainty of the task's position, if the robot adopts the method that the one who detects the task will complete the task, it will result in the imbalance of individual robot's allocation to the task. To solve this problem, we set the upper limit of the task pool capacity of each robot, namely $\lceil |\Omega_k| \,/\, |g_k| \rceil$. Simultaneously, the dynamic tasks that have been detected are filled into the task pool of each robot in turn through polling, to ensure the difference of the number of tasks in each task pool is ±1. The robot's task pool is shown in Figure 7, and the robot uses the queue to manage the task pool.

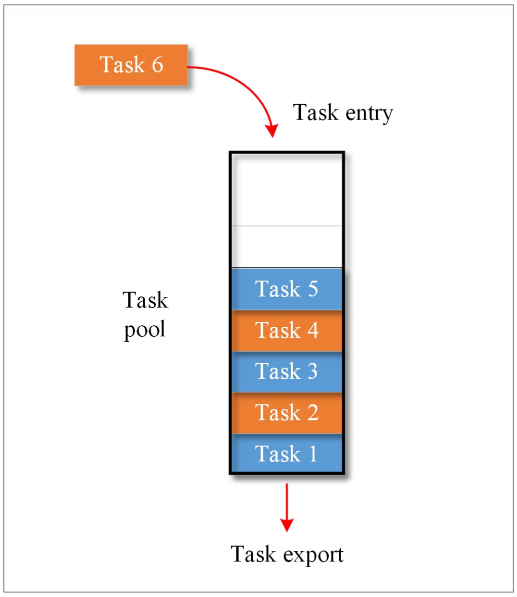


Figure 7. The task pool of robot.

The regional task balanced reallocation algorithm based on the task pool is shown in Algorithm 2. All tasks in the regional task Ω_k are added to the queue TP firstly (line 1), then the task is pulled out of the queue header and allocated to the robot with the least number of tasks in the robot group g_k (lines 3–6). Finally, all individual robots realize balanced task allocation.

Algorithm 2 Balanced reallocation algorithm for regional tasks.

```
Input:
    The robot group $g_k$ and its corresponding regional task $\Omega_k$
Output:
    The task allocation of all the robots in the group was balanced
1: InQueue(TP, $\Omega_k$);
2: while TP! $=$ $NULL$ do
3:     for $\forall \alpha \in g_k$ do
4:         task $\leftarrow$ DeQueue(TP);
5:         InQueue $(\arg\min_{\mathrm{TP}_\alpha} |\mathrm{TP}_\alpha|$, task $)$;
6:     end for
7: end while
```

6. Simulation and Experimental Results

In the experiment, we used the NetLogo simulation tool to establish the swarm robots' simulation model from task detection to task completion. We verified the effectiveness of the improved algorithm proposed in this paper by simulating the swarm robots' allocation to the task and completing the task process. The experimental operating system was Win10, and the hardware resources included 4G memory and I5 CPU processor.

6.1. Experimental Assumptions and Parameters Setting

Before presenting the simulation results, we would like to state some general assumptions in the simulations as follows:

- All robots are homogeneous and have the same capabilities and resources, and move at the same speed.
- All robots and tasks are defined as a point in two-dimensional space. Each robot can get the task point's coordinate position through perception and can calculate the distance from the task point. All coordinates are based on the local coordinate system.
- The robots communicate locally, and each robot in the system obtains messages by broadcasting.

Before the experiment, we studied the influence of robot detection distance on experimental results. In the simulated environment, 200 task points were set, and 20 robots were selected to obtain their respective task completion rate curves at different detection distances, as shown in Figure 8. When the robot's detection range is 1, 3, and 5, the task completion speed is slow. When the detection distance is greater than or equal to 7, the task completion rate curves almost coincide. Since the real robot has a specific perception range, we chose seven as the robot's detection distance to realize the simulated environment's authenticity.

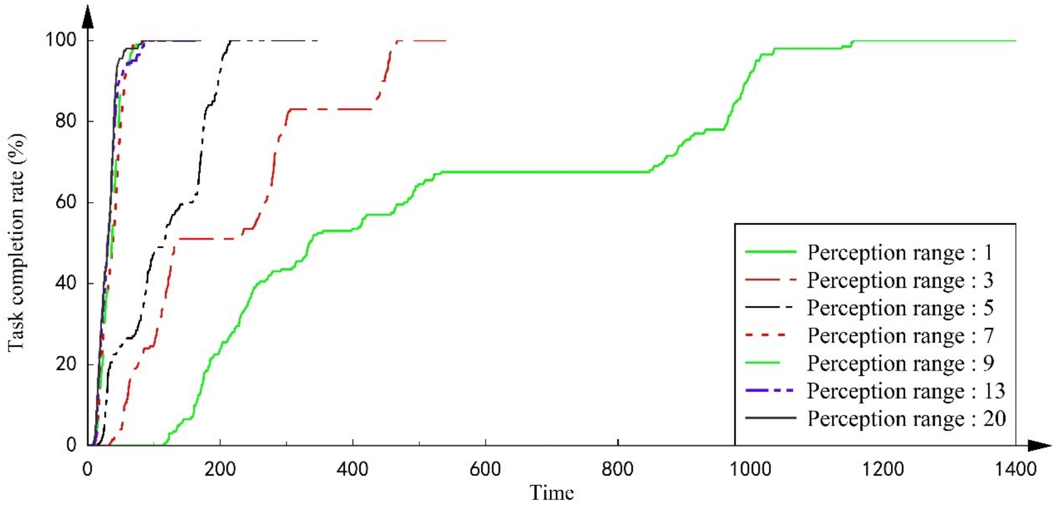


Figure 8. Influence of different perception range on task completion rate.

The experiment's task environment in this paper is two-dimensional space with 45 × 45. Through many experiments, we set the robot's communication distance as 15, which can maintain the communication connection between the swarm robots. Especially after the swarm robots are divided into several groups, the communication between them will not be able to connect.

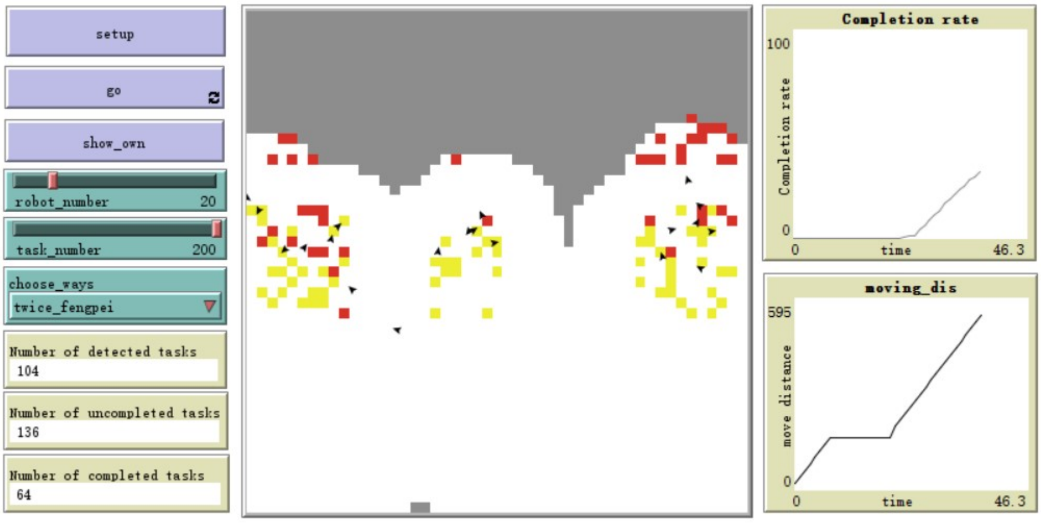
The experimental parameters are set as shown in Table 1.

Table 1. Parameter values.

Parameter	Describe	Value
d_c	Perception range	7
$\mathbf{v}$	Velocity	1
R_c	Communication range	15
θ	Distance threshold for regional task	8

6.2. Experimental Analysis

Figure 9 shows the interface of the NetLogo simulation experiment and related parameter Settings. The relevant parameters mainly include the number of robots and the number of tasks.



robot_number 20

task_number 200

choose_ways
twice_fengpei

Number of detected tasks
104

Number of uncompleted tasks
136

Number of completed tasks
64

Figure 9. Experimental interface and relevant parameter settings.

Figure 10 shows the course of the experiment. Figure 10a shows the swarm robots moving upward from the bottom to detect the task. Figure 10b shows the process of the swarm robots performing the detected task. Figure 10c shows that the swarm robots have completed all tasks.

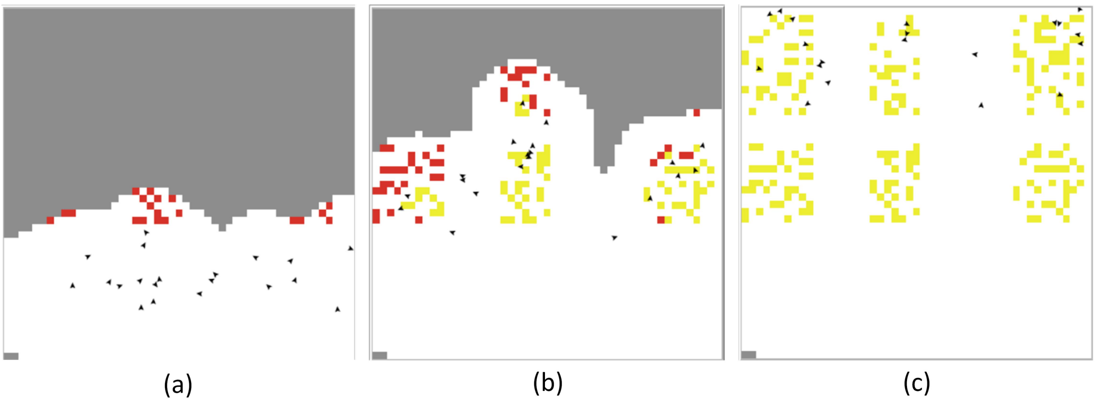


Figure 10. Demonstration diagram of experimental operation process. The black dots are the swarm robots. The shaded part is the undetected area, the white part is the detected area, and the red dot is the task point detected by the robot, and the yellow point is the completed task. (**a**) The robots begin to detect the new task. (**b**) The robots perform the tasks. (**c**) The robots complete all tasks.

In order to evaluate the performance of the method proposed in this paper, we conducted a series of task allocation experiments with different conditions and algorithms to measured and compared relevant data such as the task completion time, the task completion rate, the total movement distance of the swarm robots, and the number of tasks completed by each robot. Each experiment was run 50 times and averaged for comparison.

The task completion rate reflects the effect of task completion and is also an optimization goal considered by the task allocation algorithm in this paper. Under the condition that the task environment is roughly the same, the sooner the task completion rate reaches the stable state, the better the task completion effect will be.

6.2.1. Task Completion Rate Comparison under the Different Number of Robots

When the number of tasks remains the same at 200, we use 10, 30, 50, 70, and 100 robots to detect and perform the tasks respectively by the optimal mass transport method (OMT). Figure 11 shows the task completion rate of robots in the experiment.

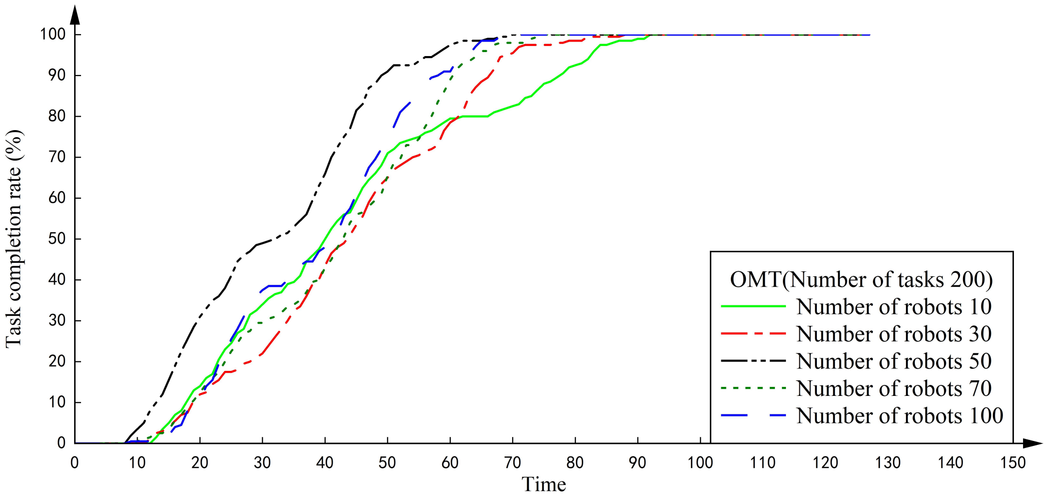


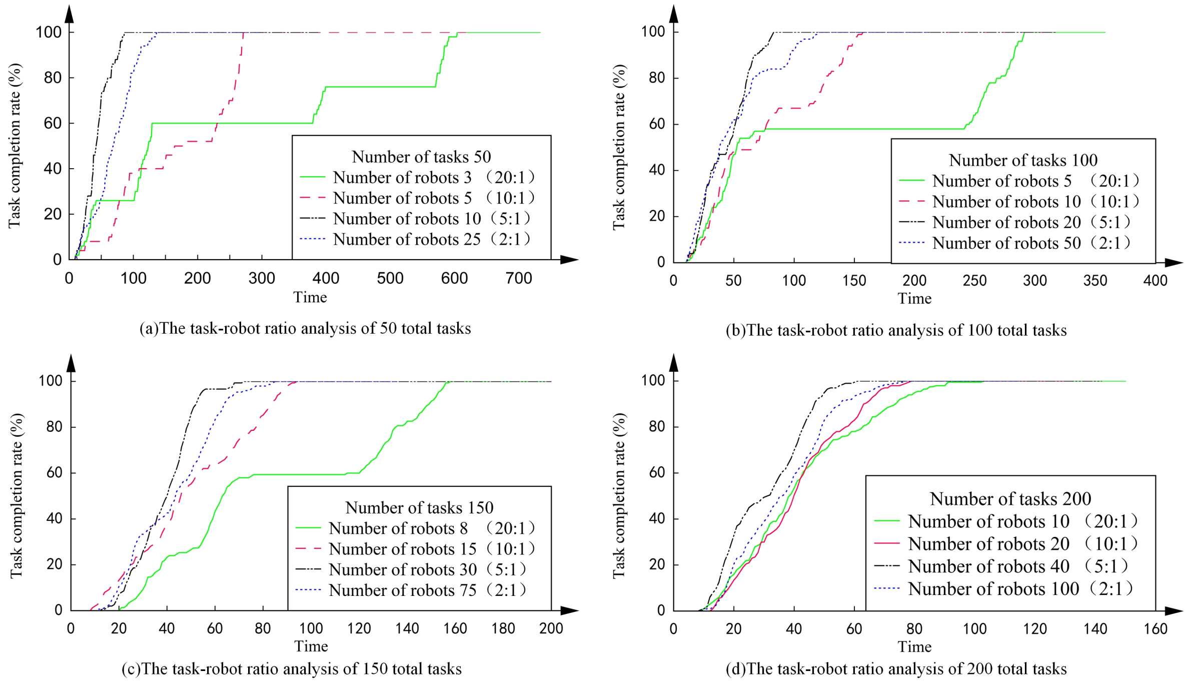
Figure 11. Comparison of the task completion rates of robots with different numbers.

In Figure 11, the time of the completion rate with 50 robots reached 100% was significantly lower than with 10 robots and 30 robots. However, when the number of robots was 70, and the number of robots was 100, the task completion time was higher than that when the number of robots was 50. Thus, when the total number of dynamic tasks remains unchanged, the task completion time does not necessarily shorten with the increase of the number of robots, so there is a good ratio relationship between the number of tasks and the number of robots. To further study the direct relationship between the number of tasks and the number of robots, we set the initial number of tasks as 50, 100, 150, and 200. Besides, the number of robots was determined by the ratio of task number to robot number of 2:1, 5:1, 10:1, and 20:1, respectively, for detection and task execution. The task completion of the robot in task execution was counted again in multiple experiments, as shown in Figure 12.

As can be seen from Figure 12, when the ratio of the number of tasks to the number of robots is 5:1, the task performance is the best. Generally speaking, when the total number of tasks is determined, the task completion time will be shorter with the increase in the number of robots. However, as the number of robots increased, so did the number of conflicts between robots, which may slow the completion of tasks. When the ratio of the number of tasks to the number of robots was 2:1, the task completion efficiency was lower than 5:1.

In Figure 12a–c, when the number of robots was too small, the robot with limited perception ability may be unable to detect the new task for a while. Then the flat curve of the task completion rate did not reach 100%. When the task completion rate is 60%, the task completion rate curve tends to be horizontal.

This experiment compares the completion speed of tasks with a different number of tasks under the same task number and machine number ratio. As the number of tasks increased, so did the number of robots, under a specific experimental environment, the more robots, the shorter the task completion time and the faster the system convergence, that is, the larger the number of tasks, the shorter the time for the task completion rate curve to reach 100%.



(a)The task-robot ratio analysis of 50 total tasks

(b)The task-robot ratio analysis of 100 total tasks

(c)The task-robot ratio analysis of 150 total tasks

(d)The task-robot ratio analysis of 200 total tasks

Figure 12. Comparison of the task completion rate with different ratios of task number and robot number.

6.2.2. Comparison of the Task Completion Rate of Different Task Allocation Methods

The comparison of relevant task allocation algorithms mainly includes the Greedy strategy [30], Hungary algorithm [31], DFS algorithm [32], the Optimal Mass Transport (OMT) method, the Optimal Mass Transport method based on adaptive grouping (OMT(a)) and the Optimal Mass Transport method based on task balanced reallocation (OMT(b)).

The number of tasks is 200, and the number of robots is 40, that is, the ratio of the number of tasks to the number of robots is 5:1. The different task allocation methods described above were used to compare the task completion rate and the task completion time. Each method was tested 50 times, and the average experimental data obtained, as shown in Figure 13.

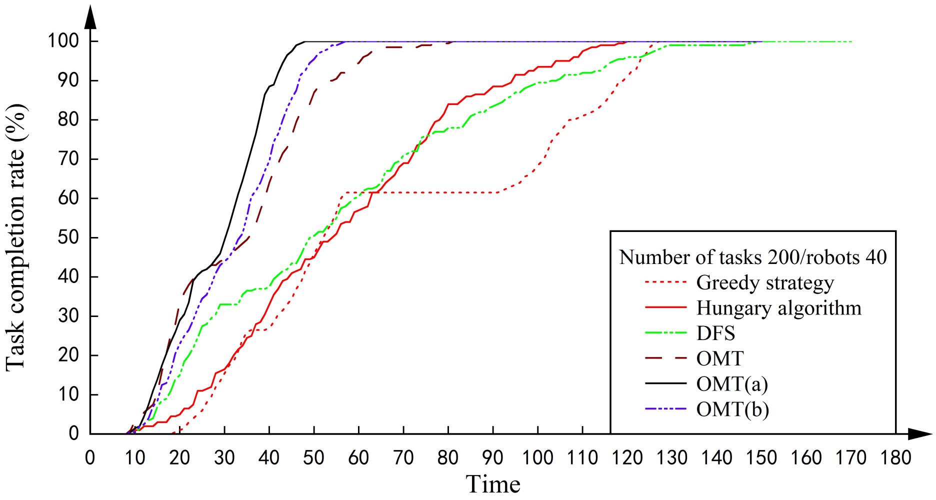


Figure 13. Comparison of the task completion rate with different task allocation methods.

Among all the experimental methods, the task completion time of the OMT method is significantly lower than that of the Greedy strategy, the Hungary algorithm, and the DFS algorithm. In the experiments, the task completion rate curve obtained by the OMT-a method has the shortest time to achieve stability. The OMT-b method considers the task balance of a single robot, and task reallocation will bring some additional computing costs, resulting in a higher task completion time than the OMT-a method.

6.2.3. A Comparison of the Total Distance Traveled by Robots with Different Task Allocation Methods

The total movement distance of the robot reflects the total movement energy consumption of the whole robot system. The smaller the movement distance, the smaller the energy consumption of the robot. Therefore, the robot's movement distance is also one of the objectives that the task allocation algorithm needs to optimize. Set the total number of tasks at 200 and the number of robots at 40. Conduct 50 experiments on the above methods, respectively, and obtain the average of the swarm robots' total movement distance for comparative analysis, as shown in Figure 14. In Figure 14, the OMT(a) method's curves and the OMT(b) method's curves almost overlap, which have the shortest completion time and the shortest total distance.

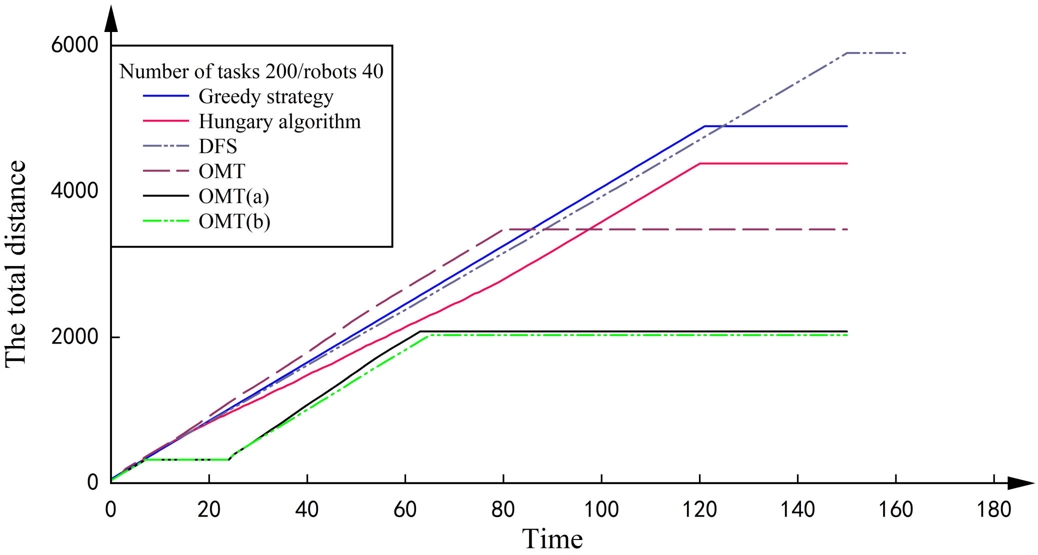


Figure 14. Comparison of the total movement distance of robots with different task allocation methods.

The OMT method directly solves how multiple suppliers can reasonably provide products to multiple demanders with cost minimization, which have less computational cost for the dynamic regional task allocation. In the quantitative comparison of experiments, the Greedy strategy, Hungary algorithm, DFS algorithm that are in the same simulation environment, compared with the OMT method proposed in this paper, the calculation cost is higher, as shown in Figures 13 and 14.

6.2.4. Comparison of the Individual Robot's Balanced Tasks under Different Task Allocation Methods

In the homogeneous swarm robot system, since there is no difference in each robot's capability and resources, the number of tasks completed by each robot needs to be roughly equal, to realize the balance of task allocation and thus improve the task completion ability of the whole system.

To avoid accidental balance in the allocation of tasks, we used 20 robots to complete 50 tasks, 100 tasks, and 200 tasks, respectively. In each case, 50 experiments were conducted. The number of tasks completed by each robot was recorded, and the average value of the 50 times was taken to compare the number of tasks allocated by individual robots, as shown in Figures 15–17.

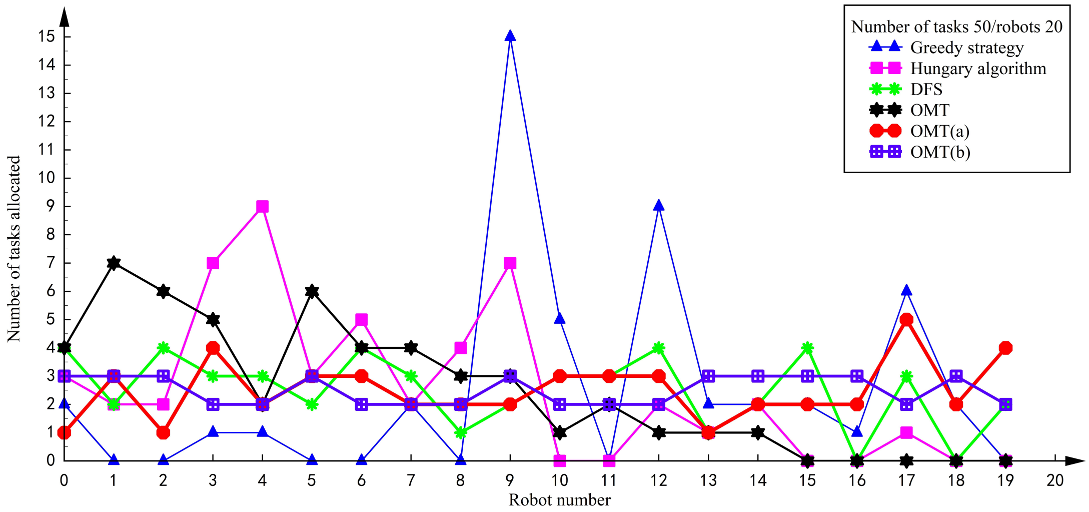


Figure 15. Comparison of the individual robot's balanced tasks for 50 tasks.

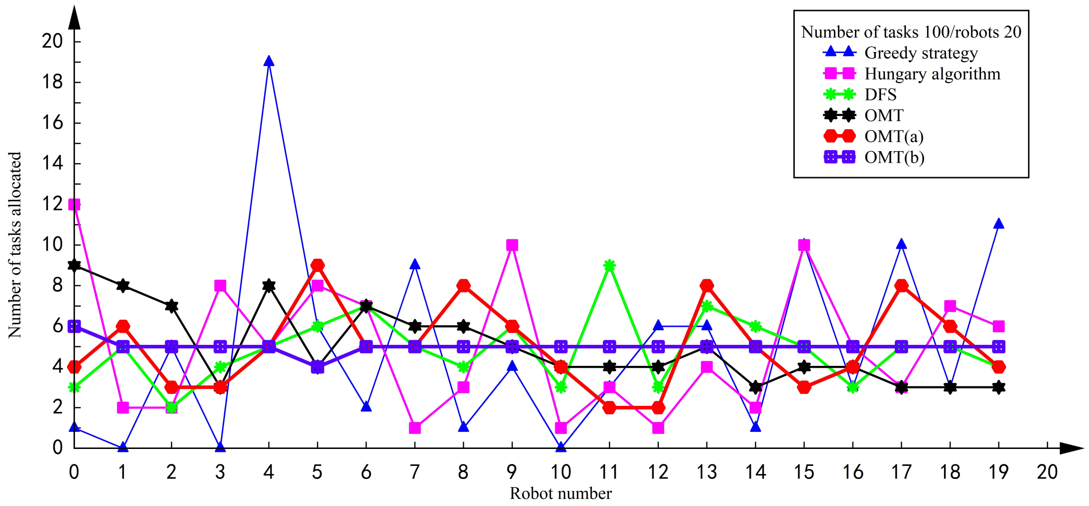


Figure 16. Comparison of the individual robot's balanced tasks for 100 tasks.

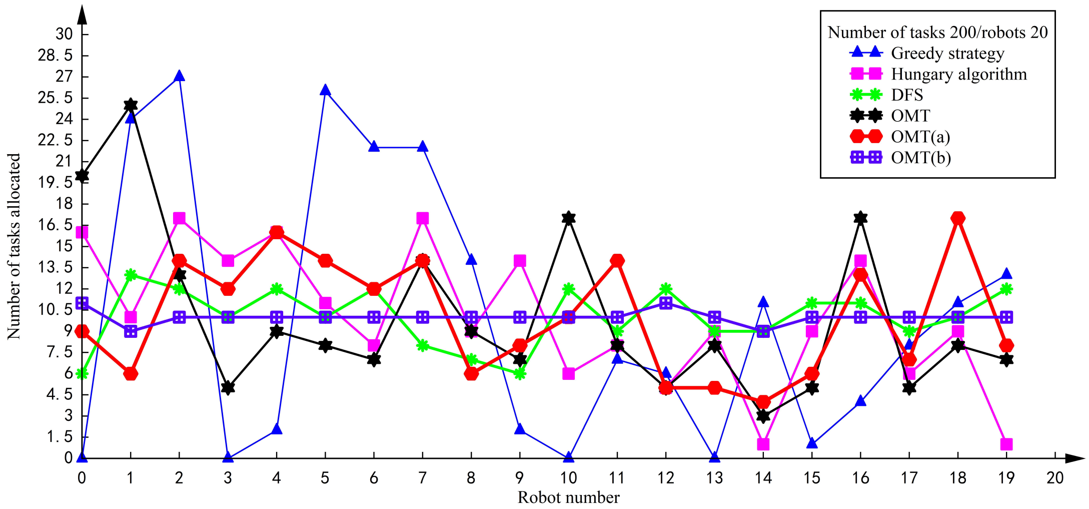


Figure 17. Comparison of the individual robot's balanced tasks for 200 tasks.

The diagrams show that the curve of the greedy strategy corresponding to the most volatile, each robot to complete the task number difference is the largest. The OMT(b) method curve is the most stable, and almost no difference between the number of each robot to complete the task reflects the method has the best balance for the task allocation of the swarm robots.

To show the task balance of different algorithms more intuitively, when the number of tasks is 200, and the number of machines is 20 and 40, the corresponding value of tasks completed by the robot in each algorithm is recorded respectively, as shown in Tables 2 and 3.

Table 2. Task balance comparison of different algorithms with 20 robots.

Method	Maximum	Minmum	Median	Mean	SD	Balance
Greedy	27	0	7.5	10	9.3	worst
Hungarian	17	1	9	10	4.7	worse
DFS	13	6	10	10	2.0	good
OMT	25	3	8	10	5.7	worse
OMT(a)	17	4	9.5	10	4.0	better
OMT(b)	11	9	10	10	0.4	best

Table 3. Task balance comparison of different algorithms with 40 robots.

Method	Maximum	Minmum	Median	Mean	SD	Balance
Greedy	23	0	1.5	5	6.4	worst
Hungarian	10	1	5	5	2.5	worse
DFS	9	1	5	5	1.9	good
OMT	13	1	3.5	5	3.0	worse
OMT(a)	10	2	5	5	2.1	better
OMT(b)	6	4	5	5	0.2	best

We have analyzed and compared the swarm robots with local perception ability for large-scale dynamic task allocation through the above series of experiments. The methods provided in this paper enable the swarm robots to complete dynamic tasks as soon as possible in a self-organizing way. The OMT(a) method has the shortest time to complete the task. The OMT(b) method's task completion time is slightly longer than that of the OMT-a method, but it has the most balanced task allocated to the individual robot.

7. Conclusions and Future Works

Task allocation methods of swarm robots are mainly divided into the centralized method and distributed method. Compared with the centralized method, the distributed method is more flexible and robust. However, for most distributed methods, they cannot be effectively applied to large-scale tasks or cannot handle dynamic tasks. For a few task allocation algorithms that can deal with large-scale tasks or dynamic tasks, there are also some problems, such as high computing cost, difficulty in adaptive adjustment to task environment change, and unbalanced task allocation. A further comparative analysis is made on various task allocation methods of swarm robots, as shown in Table 4. For large-scale dynamic tasks, our proposed methods can quickly and evenly allocate tasks.

This paper proposes a self-organizing dynamic task allocation method based on the optimal mass transport theory (OMT). We then propose an improved dynamic task allocation method based on adaptive grouping (OMT(a)) to solve the large-scale tasks. To realize individual robots' task allocation equilibrium, we further propose the optimal mass transport method based on task balanced reallocation (OMT(b)). Experiments show the effectiveness of the proposed methods. Future work will include experiments with real robots, implementing complex task models, and solving collaborative task allocation problems.

Table 4. Comparison of task allocation methods for the swarm robots.

Method	Distributed	Adaptive Adjustment	Large-Scale Task	Dynamic Task Dynamic Task	Balanced Allocation	Task Completion Time	Computing Cost
GA [10]	×	×	√	×	×	short	large
AC [11]	×	√	×	×	√	short	small
DTS [12]	×	×	√	×	×	long	large
GM [13]	×	×	×	×	×	long	large
DA [14]	√	√	×	×	×	long	small
SOM-NN [17]	√	√	×	√	×	short	small
Greedy [30]	√	√	√	√	×	long	small
Hungarian [31]	×	×	×	×	×	long	large
DFS [32]	×	×	×	√	×	long	large
OMT	√	√	√	√	×	short	small
OMT(a)	√	√	√	√	×	very short	small
OMT(b)	√	√	√	√	√	very short	small

Author Contributions: Conceptualization, supervision and project administration, X.M.; methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was financially supported by the Natural Science Foundation of China under grants 61532004 and 2018YFB1004202.

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

References

1. Şahin, E. Swarm Robotics: From Sources of Inspiration to Domains of Application. In *International Workshop on Swarm Robotics*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; pp. 10–20.
2. Bayındır, L. A Review of Swarm Robotics Tasks. *Neurocomputing* **2016**, *172*, 292–321. [CrossRef]
3. Sun, Q.; Diao, M.; Zhang, Y.; Li, Y. Cooperative Localization Algorithm for Multiple Mobile Robot System in Indoor Environment Based on Variance Component Estimation. *Symmetry* **2017**, *9*, 94. [CrossRef]
4. Garcia-Aunon, P.; Barrientos Cruz, A. Comparison of Heuristic Algorithms in Discrete Search and Surveillance Tasks Using Aerial Swarms. *Appl. Sci.* **2018**, *8*, 711. [CrossRef]
5. Mellinger, D.; Shomin, M.; Michael, N.; Kumar, V. Cooperative Grasping and Transport Using Multiple Quadrotors. In *Distributed Autonomous Robotic Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 545–558.
6. Beck, Z.; Teacy, W.L.T.; Jennings, N.R.; Rogers, A.C. Online Planning for Collaborative Search and Rescue by Heterogeneous Robot Teams. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, 9–13 May 2016; pp. 1024–1033.
7. Innocente, M.S.; Grasso, P. Self-Organising Swarms of Firefighting Drones: Harnessing the Power of Collective Intelligence in Decentralised Multi-Robot Systems. *J. Comput. Sci.* **2019**, *34*, 80–101. [CrossRef]
8. Hu, J.; Xu, J.; Xie, L. Cooperative Search and Exploration in Robotic Networks. *Unmanned Syst.* **2013**, *1*, 121–142. [CrossRef]
9. Khaldi, B.; Cherif, F. An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-Robotics. *Int. J. Comput. Appl.* **2015**, *126*, 31–37. [CrossRef]
10. Shima, T.; Rasmussen, S.J.; Sparks, A.G.; Passino, K.M. Multiple Task Assignments for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms. *Comput. Oper. Res.* **2006**, *33*, 3252–3269. [CrossRef]
11. Zheng, T.; Yang, L. Optimal ant colony algorithm based multirobot task allocation and processing sequence scheduling. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation (WCICA), Chongqing, China, 25–27 June 2008; pp. 5687–5692.
12. Higgins, A.J. A Dynamic Tabu Search for Large-Scale Generalised Assignment Problems. *Comput. Oper. Res.* **2001**, *28*, 1039–1048. [CrossRef]
13. Ghassemi, P.; Chowdhury, S. Decentralized Task Allocation in Multi-Robot Systems via Bipartite Graph Matching Augmented With Fuzzy Clustering. In Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, QC, Canada, 26–29 August 2018.

14. Liang, J. Research on Distributed Task Allocation Algorithm for Unmanned Aerial Vehicles Based on Consensus Theory. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 4892–4897.
15. Das, G.P.; McGinnity, T.M.; Coleman, S.A.; Behera, L. A Distributed Task Allocation Algorithm for a Multi-Robot System in Healthcare Facilities. *J. Intell. Robot. Syst.* **2015**, *80*, 33–58. [CrossRef]
16. Wang, L.; Wang, Z.; Hu, S.; Liu, L. Ant Colony Optimization for Task Allocation in Multi-Agent Systems. *China Commun.* **2013**, *10*, 125–132. [CrossRef]
17. Zhu, A.; Yang, S.X. A Neural Network Approach to Dynamic Task Assignment of Multirobots. *IEEE Trans. Neural Netw.* **2006**, *17*, 1278–1287. [PubMed]
18. de Oca, M.A.M.; Ferrante, E.; Scheidler, A.; Pinciroli, C.; Birattari, M.; Dorigo, M. Majority-Rule Opinion Dynamics with Differential Latency: A Mechanism for Self-Organized Collective Decision-Making. *Swarm Intell.* **2011**, *5*, 305–327. [CrossRef]
19. De Mendonça, R.M.; Nedjah, N.; de Macedo, Mourelle, L. Efficient Distributed Algorithm of Dynamic Task Assignment for Swarm Robotics. *Neurocomputing* **2016**, *172*, 345–355. [CrossRef]
20. Brutschy, A.; Pini, G.; Pinciroli, C.; Birattari, M.; Dorigo, M. Self-Organized Task Allocation to Sequentially Interdependent Tasks in Swarm Robotics. *Auton. Agents Multi-Agent Syst.* **2014**, *28*, 101–125. [CrossRef]
21. Ghassemi, P.; DePauw, D.; Chowdhury, S. Decentralized Dynamic Task Allocation in Swarm Robotic Systems for Disaster Response. *arXiv* **2019**, arXiv:1907.04394.
22. Garapati, K.; Roldán, J.J.; Garzón, M.; del Cerro, J.; Barrientos, A. A Game of Drones: Game Theoretic Approaches for Multi-Robot Task Allocation in Security Missions. In Proceedings of the Iberian Robotics Conference, Sevilla, Spain, 22–24 November 2017; pp. 855–866.
23. Haker, S.; Zhu, L.; Tannenbaum, A.; Angenent, S. Optimal Mass Transport for Registration and Warping. *Int. J. Comput. Vis.* **2004**, *60*, 225–240. [CrossRef]
24. Rachev, S.T. and Rüschendorf, L. *Mass Transportation Problems: Volume I: Theory*; Springer Science & Business Media: New York, NY, USA, 1998; Volume 1.
25. Patel, R.; Frasca, P.; Bullo, F. Centroidal area-constrained partitioning for robotic networks. *J. Dyn. Syst. Meas. Control.* **2014**, *136*, 031024. [CrossRef]
26. Cortes J. Coverage Optimization and Spatial Load Balancing by Robotic Sensor Networks. *IEEE Trans. Autom. Control.* **2010**, *55*, 749–754. [CrossRef]
27. Arslan, O.; Koditschek, D.E. Voronoi-based coverage control of heterogeneous disk-shaped robots. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4259–4266.
28. Dong, S.; Xu, K.; Zhou, Q.; Tagliasacchi, A.; Xin, S.; Nießner, M.; Chen, B. Multi-robot collaborative dense scene reconstruction. *ACM Trans. Graph.* **2019**, *38*, 1–16. [CrossRef]
29. Wang, Q.; Mao, X.; Yang, S.; Chen, Y.; Liu, X. Grouping-based adaptive spatial formation of swarm robots in a dynamic environment. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418782359. [CrossRef]
30. Yamauchi, B. Decentralized coordination for multirobot exploration. *Robot. Auton. Syst.* **1999**, *29*, 111–118. [CrossRef]
31. Giordani, S.; Lujak, M.; Martinelli, F. A Distributed Algorithm for the Multi-Robot Task Allocation Problem. In Proceedings of the 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems-Volume Part I, Córdoba, Spain, 1–4 June 2010; Volume 6096, pp. 721–730.
32. Zu, L.; Tian, Y.; Liu, J. Algorithms of Task-Allocation and Cooperation in Multi Mobile Robot System. In Proceedings of the Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788), Hangzhou, China, 15–19 June 2004; IEEE: Piscataway, NJ, USA, 2004; Volume 4, pp. 2841–2845.

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