Preprints.org

Article

Dynamic Response Threshold Model for Self-organized Task Allocation in a Swarm of Foraging Robots

Bao Pang , Ziqi Zhang , Yong Song* * , Xianfeng Yuan , Qingyang Xu

Article

Dynamic Response Threshold Model for Self-Organized Task Allocation in a Swarm of Foraging Robots

Bao Pang [1], Ziqi Zhang [1], Yong Song [1,*], Xianfeng Yuan [1,2,3] and Qingyang Xu [1]

[1] School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China; pangbao@sdu.edu.cn (B.P.); 202237596@mail.sdu.edu.cn (Z.Z.); songyong@sdu.edu.cn (Y.S.); yuanxianfeng@sdu.edu.cn (X.Y.); qingyangxu@sdu.edu.cn (Q.X.)

[2] Shandong Jinpiao Food Machinery Co., Ltd, Weihai 264209, China

[3] Institute of Oceanology, Chinese Academy of Sciences，Qingdao 266071, China

* Correspondence: songyong@sdu.edu.cn

Abstract: In swarm-robotics foraging, the purpose of task allocation is to adjust the number of active foraging robots dynamically based on the task demands and changing environment. It is a difficult challenge to generate self-organized foraging behavior in which each robot can adapt to environmental changes. To complete the foraging task efficiently, this paper presents a novel self-organized task allocation strategy known as the dynamic response threshold model (DRTM). To adjust the behavior of the active foraging robots, the proposed DRTM newly introduces the traffic flow density, which can be used to evaluate the robot density. First, the traffic flow density and the amount of obstacle avoidance are used to adjust the threshold which determines the tendency of a robot to respond to a stimulus in the environment. Then, each individual robot uses the threshold and external stimulus to calculate the foraging probability that determines whether or not to go foraging. Simulation and physical experiments are carried out to evaluate the performance of the proposed method. The experimental results show that the DRTM is superior to and more efficient than the adaptive response threshold model (ARTM) in swarm foraging.

Keywords: swarm robotics; adaptive foraging; self-organized; task allocation; dynamical response threshold model; traffic flow density

1. Introduction

Inspired by the foraging of ant colonies, numerous studies [1–3] have used activation-threshold model to implement task allocation in a swarm of foraging robots. Although such mechanisms generate efficient task allocation (TA), some drawbacks remain. With a higher threshold, an individual robot is apt to stay in the nest; with a lower threshold, having many robots forage simultaneously increases the amount of physical interference [4]. Therefore, a swarm with a fixed threshold lacks adaptive capacity and cannot respond properly to an uncertain and dynamic environment. Moreover, as has been noted elsewhere, the performance of swarm-robotics foraging is influenced by the physical interference among the active foraging robots [5,6]: having too many robots foraging simultaneously leads to less-efficient foraging [7,8]. However, previous studies on TA did not take physical interference into account when adjusting the number of active foraging robots [9–11].

To overcome the above drawbacks, this paper proposes the dynamic response threshold model (DRTM) with the aim of achieving self-organized TA and completing foraging tasks efficiently. First, our approach uses a mutative threshold that varies with the external environment (e.g., the number of active foraging robots and the number of times robots avoid the obstacles), so DRTM makes the swarm system adaptive to the changing environment and robust to failures. Second, we also newly take into account the physical interference and traffic conditions to adjust the behavior of the robots

dynamically. Clearly, the greater the density of active foraging robots in the environment, the higher the amount of physical interference. Therefore, to adjust the number of active foraging robots, we newly propose the concept of traffic flow density (TFD), which denotes the density of active foraging robots. To better describe the traffic conditions, the DRTM also uses the amount of physical interference to adjust the robots' behavior. In the proposed approach, the TFD and the amount of interference are both used to adjust the response threshold dynamically. When there is traffic congestion in the foraging area, the threshold rises and the swarm allocates fewer robots to foraging tasks; when there is no traffic congestion, many more robots are sent foraging. Therefore, by considering traffic information from the environment, the DRTM can reduce the amount of physical interference among robots and improve the foraging efficiency.

The remainder of this paper is organized as follows. In Sect. 2, we briefly review existing studies in the field of TA. In Sect. 3, we describe the attractor selection model (ASM) and DRTM used in swarm foraging and briefly introduce the simulation setup and performance measures. In Sect. 4, we present the results of experiments to demonstrate the effectiveness of the proposed method. In Sect. 5, we explore swarm foraging further under different experimental conditions. In Sect. 6, we end the paper with some concluding remarks.

2. Background

Nature is full of examples of social insects that cooperate efficiently to complete complex tasks. A typical example is the foraging done by ant colonies, in which the ants search for food in their environment and transport it to their nest [12]. Inspired by such foraging behavior, various studies [13–16] have used simple robots to complete complex foraging tasks. In such tasks, the robots search for certain objects (known as food tokens) that are scattered in the foraging arena. Once a food token is found, the robot captures it and delivers it to a specific area designated as the nest [17]. Swarm foraging can be treated as a comprehensive process of searching, collecting, and transferring. In the future, it could be used to clean up toxic waste, collect samples in unknown regions, clear land mines, and for search and rescue and planetary exploration [18,19].

Most previous studies of swarm foraging analyzed how the swarm energy changes. Each food token contains a certain amount of energy that increases the swarm energy upon being collected by a robot. However, some swarm energy is lost to the movement of the active foraging robots and the interference among them (e.g., blocking the paths of others or collisions that slow the robots down). The swarm's nest can also entail energy being lost at a certain rate (Crate). To maintain its energy at the desired level, the swarm must implement self-organized TA: the control algorithm of each robot must decide when the robot should search for food tokens and when it should remain in the nest. Such TA can be observed in social insects such as ants [20]. In an ant colony, when there are fewer food tokens (stimuli) than a certain threshold θ, many more ants go foraging; when there are more food tokens than this threshold, fewer ants go foraging [21]. Bonabeau et al. proposed a simple activation-threshold model to account for the experimental observations of how an ant colony divides its labor [22].

In a swarm of foraging robots, TA refers to the ability to adjust the number of active foraging robots dynamically based on local decisions made by each robot. Some TA mechanisms are based only on threshold values, whereby robots decide to go foraging when an external stimulus exceeds a threshold value. The external stimulus could be related to the number of food tokens stored in the nest or the amount of time spent in each state by each robot. To maintain the nest energy at the desired level, Kreiger & Billeter implemented a simple TA mechanism based on individual activation thresholds that are fixed and distributed equally among 75 − 100% of the desired nest energy [1]. When the nest energy is less than the threshold value, the robots begin to search for food token. Agassounon & Martinoli dealt specifically with the advantages and drawbacks of using threshold-based TA for foraging tasks [23]. Liu et al. presented a simple adaptation mechanism to adjust the ratio of foragers to resters automatically and adjusted the threshold values according to internal, environmental, and social cues [24]. When the time spent in each state exceeds the corresponding

threshold, the individual robot switches to the next state. Using threshold-based methods, many robots forage simultaneously, which can lead to undesired physical interference.

Other TA mechanisms are based on probabilistic methods, wherein the threshold values and other information are used to calculate probability. In a probabilistic method, the decision of each robot to switch its activity is taken based on a probability. A simple adaption rule is then used to adjust the probability of a robot leaving the nest; the probability is raised or lowered as a fixed constant multiplied by the number of consecutive successes or failures at collecting food tokens [10]. Buchanan et al. [25] and Brutschy et al. [26] presented a self-organized method to partition a task into sequentially interdependent subtasks based on probabilistic methods. Pitonakova et al. [27] achieved self-organized TA based on a probabilistic method, wherein at each time step an observer can become a scout with a fixed probability. In such probabilistic methods, various factors (e.g., information about food and the environment) that could affect the foraging efficiency are not accounted for. Therefore, when researching TA, many studies have used probability methods based on response threshold model. Bonabeau et al. introduced a fixed response threshold model (FRTM) to account for experimental observations of how social insects divide their labor [22,28]. However, the FRTM uses a fixed threshold and so cannot cope with uncertain and dynamic environments. Therefore, Lee et al. [29] and Lope et al. [30] used an response threshold model with a variable threshold; when a robot engages in a task, the task-associated threshold is decreased. Kanakia et al. proposed a sigmoid response threshold function to respond to changing task requirements by adapting the static response thresholds [31]. Castello et al. proposed an extended FRTM known as ARTM using a discretized version of the attractor selection paradigm to control the threshold parameters dynamically [9,11]; when the stimulus decreases, the threshold tends to increase and fewer robots go foraging; when the stimulus increases, the threshold tends to decreases and far more robots go foraging. In ARTM, the changes regarding the threshold are only related to the number of food tokens in nest. Kanakia et al. not only provided a theoretical basis for response threshold functions of agents that share information being optimal strategies but also reinforced that using adaptive response thresholds is the correct strategy for the type of scenario investigated herein [32]. However, although response threshold model improve foraging efficiency, they do not consider physical interference when seeking to improve TA performance.

3. Methods

In swarm foraging, a robotic system is homogeneous. All robots follow the same behavior rules when foraging, as shown in Figure 1. The states for swarm foraging are as follows:
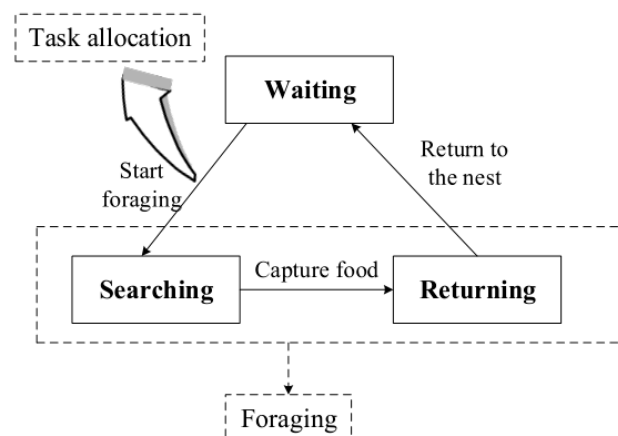

Figure 1. State transitions in foraging. An individual robot switches its behavior among the given three states in response to a specific external stimulus. The states of searching and returning are foraging states in which the robot consumes energy. Task allocation (TA) is the process of converting from the waiting state to the searching state.

Waiting (S_W): At the outset of foraging, all robots are waiting in the nest. An individual robot initializes a timer at $t_1 = 0$ to denote its waiting time in the nest, with the minimum waiting time T_1 being given in advance. When the waiting time expires ($t_1 \geqslant T_1$), the individual robot uses the DRTM to calculate its foraging probability Ps. Based on Ps, the robot determines whether to switch from the S_W state to the S_S state.

Searching (S_S): When a robot begins foraging, the ASM controls its foraging behavior. When the robot captures a food token, it switches to the S_R state.

Returning (S_R): In the S_R state, the ASM controls the returning behavior. When the robot arrives at its nest, it deposits the food token and then switches to the S_W state.

To maintain the number of food tokens at the desired level, the swarm should use effective self-organized TA to respond to its changing environment. We employ a control strategy based on the ASM to control the movement of each robot. Herein, we propose the DRTM to generate self-organized TA to adjust the number of foraging robots dynamically.

3.1. Attractor Selection Model (ASM)

Inspired by the dynamic behavior of biomolecules that can adapt to a new, unknown, and noisy environment [33], Kashiwagi et al. proposed a simple and robust ASM to respond robustly to unknown environmental changes without requiring a large number of specific sensors and transducers [34].

The ASM is simple and does not require a model of the environment. It is therefore robust to unknown environmental changes. Therefore, we propose a method for controlling a robot's movement based on the ASM. The ASM is described by the following Langevin equation:

$$\dot{x} = f(x) \times A(t) + \varepsilon \tag{1}$$

where x and $f(x)$ represent the state and the dynamics of the ASM, respectively. The function $f(x)$ can be designed to have some attractors, and ε is a noise term. The term $A(t)$ is the called 'activity' variable and indicates the fitness of the current state to the environment. To apply the ASM to robot control, we interpret the state x as the posture (position and orientation) of the robot, therefore $\dot{x}$ represents the motion of the robot. The function $f(x)$ is designed to have some attractors that are related to particular motions. In swarm foraging, when the robot is in state S_S, the attractors are the food; when the robot is in state S_R, the attractor is the light within the nest. When the foraging robot discovers an attractor, it keeps advancing to approach the corresponding attractor. The activity $A(t)$ can also be considered as the ASM's input variable, with $A(t) = \{0, 1\}$. When the robot detects an attractor, $A(t)$ is set to 1; at this point, $f(x) \times A(t)$ controls the motion of the robot, and the robot approaches the attractor. If the robot detects no attractors, $A(t)$ is set to 0; ε controls the motion of the robot, and the robot performs a random walk to search for attractors.

As given in Table 1, at the outset of state S_S the fitness of the current state to the environment is $A(t) = 0$; the robot's motion is controlled by the noise ε and the robot performs a random walk to search for food. When the robot detects a food token, the activity is set to $A(t) = 1$; the robot's motion is then controlled by $f(x) \times A(t)$ and the robot advances to approach the food token. After capturing the food token, the robot switches to state S_R. A light source is set up in the nest, and the robot returns to the nest by tracking the light. In state S_R, the robot walks randomly to detect the light with $A(t) = 0$. When the robot detects the light, the activity is set to $A(t) = 1$; $f(x) \times A(t)$ then controls the robot's motion and the robot advances to approach the light. After depositing the food token in the nest, the robot switches to state S_W.

Table 1. Foraging behaviors based on attractor selection model(ASM).

State	Detect attractors	ASM's input: A(t)	Robot's motion
S_W	No	0	random walk
S_W	Yes	1	approach to the food
S_W	No	0	random walk
S_W	Yes	1	approach to the light

3.2. Dynamical Response Threshold Model (DRTM)

To complete the foraging task, we should maintain the swarm energy at the desired level. The food tokens collected by the foraging robots can increase the swarm energy. However, the foraging activities (e.g., the movements of the foraging robots) and interference among robots constitute a loss of swarm energy. The swarm's nest can also constitute a loss of energy at a certain rate (C_{rate}). Therefore, the amount of food (or energy) in the nest will decrease gradually, and when there are fewer food tokens than the desired level, the robots tend to go foraging based on the DRTM. By adjusting the number of foraging robots, the DRTM can complete effective TA and improve foraging efficiency. The DRTM accounts for the TFD and the amount of obstacle avoidance to adjust the response threshold dynamically. In the DRTM, the foraging behavior of an individual robot is influenced by the external stimulus ($S(t)$) and the response threshold (θ).

The external stimulus $S(t)$ reflects the amount of food in the nest and is defined as

$$S(t) = F_d - F(t) \tag{2}$$

where F_d is the desired number of food tokens to be maintained in the nest and $F(t)$ is the number of food tokens in the nest at time t. The stimulus $S(t)$ measures the gap between the existing number of food tokens and the desired number. Larger $S(t)$ means far fewer existing food tokens than desired, and therefore many more robots will be allocated to foraging. Conversely, with smaller $S(t)$, fewer robots tend to go foraging. When $S(t) \leq 0$, there are the desired number of food tokens and the robots stop going foraging.

In swarm foraging, physical interference refers mainly to obstacle avoidance. If a foraging robot detects an obstacle (e.g., another robot or a wall), it executes avoidance behavior, namely turning away from the obstacle. Such avoidance behavior may interrupt the state of the robot, and after resolving a collision, the robot may spend much time reentering its previous state. For example, if a robot advances toward its nest in a straight line by tracking the light, obstacle avoidance can cause the robot deviate from its route. After resolving the collision, the robot searches for the light once again in random walk, thereby taking longer to return to the nest. Therefore, obstacle avoidance consumes more energy and decreases foraging efficiency. To have fewer obstacle avoidance events (OAEs), we propose two measurements to evaluate the traffic conditions upon which the number of active foraging robots is adjusted dynamically. One is the TFD and the other is the average number of OAEs.

In swarm foraging, the number of OAEs is related not only to the number of active foraging robots but also to the size of the foraging arena. To assess quantitatively how the number of foraging robots and the size of the foraging arena impact the foraging task, we newly introduce the concept of TFD. As a basic concept in traffic engineering, the TFD denotes the number of vehicles within one or several lanes on a unit length of road [35] and is defined as

$$k = \frac{N_V}{L} \tag{3}$$

where N_V is the number of vehicles on the road and L is the length of the road. The TFD can reflect the traffic conditions directly and has been widely used in traffic engineering. When k is large, there is traffic congestion on the road; otherwise, the traffic is flowing smoothly. Based on the size of the TFD, different traffic management and control measures are adopted to regulate the road traffic to improve traffic efficiency.

Herein, the TFD is newly used to calculate the density of foraging robots. The logical extension of Equation (3) is

$$k = \frac{1}{L \times L} \times (N_r - N_W) \tag{4}$$

where L is the side length of the foraging arena (in this foraging scenario, robots roam a square area, so the amount of available space is $L \times L$), N_r is the total number of robots in the swarm foraging, and N_W is the number of robots in state S_W. Therefore, N_r–N_W denotes the number of active foraging robots. In swarm foraging, the TFD is the number of active foraging robots per unit area and can be used to reflect the traffic conditions. As can be seen from Equation (4), when k is large, many robots are

foraging. Much more obstacle avoidance will be generated, thereby reducing the foraging efficiency, meaning that we should decrease the number of active foraging robots.

The average number of OAEs M_A within a period of time T_2 can also reflect the traffic situation, and M_A is defined as

$$M_A = \frac{M_{T_2}}{N_r} \tag{5}$$

where M_{T2} is the total number of OAEs over time T_2. When there is traffic congestion, more avoidance behavior arises; when the traffic is moving smoothly, less avoidance behavior arises. However, traffic conditions change with time, and the number of OAEs reflects the traffic conditions over time T_2. Therefore, we need to choose a suitable time period over which to calculate the number of OAEs. Too big or too small a time period T_2 will not reflect the traffic conditions correctly. The number of OAEs due to robots within state S_R can well reflect the traffic conditions. Any robot in state S_R can return to the nest in $L/2v$ seconds, where v is the robot's speed and $L/2$ is half the side length of the foraging arena; therefore, we set $T_2 = L/2v$. The greater M_A, the heavier the traffic congestion, meaning that we should decrease the number of active foraging robots.

In the DRTM, the response threshold changes dynamically and is defined as

$$\theta = \alpha * k + \beta * M_A \tag{6}$$

where α and β are adjustment factors used to regulate the contributions of k and M_A, respectively. The quantity θ can reflect the traffic conditions. As can be seen from Equation (6), k and M_A together adjust the value of threshold θ. Larger θ means traffic congestion in the foraging arena, and therefore fewer robots should be allocated to foraging. Conversely, smaller θ means that many more robots should go foraging.

In the DRTM, the state transition probability P_s, which is the probability of leaving the nest to perform foraging, is defined as

$$P_s = \begin{cases} \dfrac{S(t)^n}{S(t)^n + \theta^n}, & S(t) > 0 \\ 0, & S(t) \le 0 \end{cases} \tag{7}$$

where n determines the slope of the probability function; different values of n may produce different responses for the same threshold θ and stimulus $S(t)$. The value of n is set randomly for each robot at the outset of foraging to avoid all robots working simultaneously. The larger the stimulus $S(t)$, the less food there is in the nest for a given desired amount of food; under such circumstances, the foraging probability P_s increases. Conversely, P_s decreases when $S(t)$ is smaller. When $S(t) \le 0$, there is the desired number F_d of food tokens in the nest and the robots can stop foraging. The traffic conditions can also affect the foraging probability; P_s decreases when traffic congestion occurs and increases when the traffic is flowing smoothly. As can be seen from Equation (7), the food information $S(t)$ and environment information θ together adjust the foraging probability, and therefore the swarm allocates a differing number of robots to go foraging based on the foraging probability. Therefore, the DRTM can not only reduce the amount of physical interference among the robots but also ensure that the foraging task can be completed.

Figure 2 shows probability response curves for various values of θ and n. According to Figure 2, when the threshold θ is equal to the stimulus $S(t)$, the foraging probability is 0.5. Therefore, the robot with the lowest θ responds fastest to the external stimulus. A robot with a higher threshold θ is less sensitive to the external stimulus and tends to wait for longer in the nest. As shown in Figure 2, n determines the speed at which the robot responds to changes in the stimulus. Robots with larger n have a steeper curve that will increase their foraging probability more abruptly when the external stimulus $S(t)$ increases.
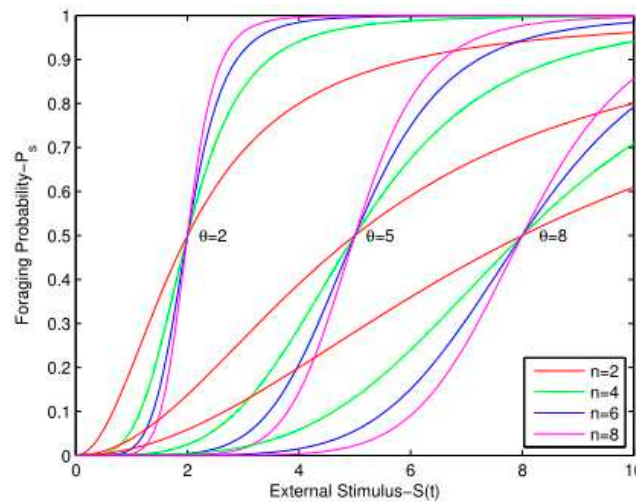
Figure 2. Dynamic response curves for various values of θ and n. The probability response curve is steeper with larger n. A robot with larger θ goes foraging when the external stimulus is larger.

Figure 3 shows a flowchart of DRTM-based swarm foraging. Robots in state S_W use the DRTM to calculate their foraging probability P_s. Based on P_s and roulette wheel selection, an individual robot determines whether or not to go foraging. If the robot goes foraging, its foraging behavior is controlled by the ASM as described in Sect. 3.1.

Figure 3. Flowchart of swarm foraging based on dynamic response threshold model (DRTM).

3.3. Simulation Setup

To evaluate the proposed model, we ran simulation experiments on the ARGoS robotic simulator [36]. Figure 4 shows a screenshot of the ARGoS simulator at the start of a simulation experiment. The simulation environment was a square area of L meters ($3 \leq L \leq 4$), and the grey circular region represents the nest with a diameter of d meters ($1 \leq d \leq 2$). The yellow object in the middle of the simulation area is a light source that allows the robots to return to the nest by sensing the intensity and direction of the light. The blue objects are robots and are placed in the nest randomly before each experiment. The black objects are food tokens and are distributed randomly in the foraging arena.

Whenever a robot delivers a food token back to the nest, a new food token is generated randomly in the foraging arena.
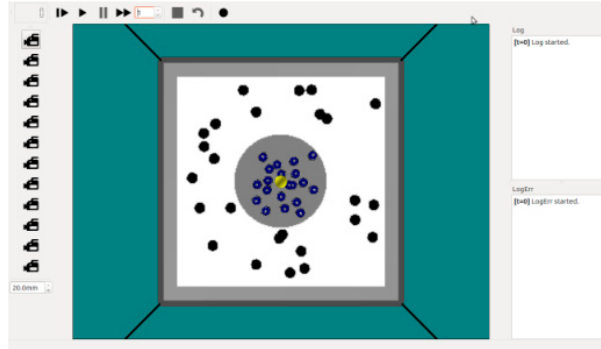

Figure 4. Screenshot of ARGoS simulator at start of simulation experiment. The yellow object is the light source, blue objects are robots, and black objects are food.

3.3. Performance Measures

This subsection describes three performance indexes used to evaluate the effectiveness of the proposed method. The purpose of swarm foraging is to maintain the food in the nest at the desired level, therefore the average deviation of food (ADF) in the nest (V_f) is an important index with which to measure the performance. It is defined as

$$V_f = \sum_{t=1}^{T} \frac{\left|F_d - F(t)\right|}{T} \tag{8}$$

where F_d is the desired number of food tokens and $F(t)$ is the number of net food tokens (the food collected by the swarm minus the food consumed by the swarm) at time t in the nest. Measure V_f denotes the gap between the existing number of food tokens in the nest and the desired number. As such, V_f can be used to assess both the robots' abilities to adapt to changes in the external stimulus and the effect of TA. Smaller V_f not only indicates that the swarm can maintain a food level that is closer to the desired value F_d but also proves that the TA is more able to adapt to changes in the food supply.

During a mission, the robots' movements, the OAEs, and the nest all consume energy that comes from the food collected by the robots. Therefore, the energy efficiency (E_e) is introduced as a performance measure. It is defined as

$$E_e = \frac{E_n}{E_T} \tag{9}$$

where E_n is the net food energy and E_T is the total energy collected by the robots. Smaller E_e implies that the swarm consumes more energy and that the TA leads to less-efficient foraging.

To reduce the amount of physical interference among the robots, we use the TFD and the number of OAEs to adjust the value of threshold θ dynamically. Therefore, the average number of OAEs M_v is also considered as a performance measure:

$$M_v = \frac{M_T}{N_r} \tag{10}$$

where M_T is the total number of OAEs during foraging and N_r is the total number of robots engaged in foraging. Because obstacle avoidance consumes energy and causes traffic congestion, M_v reflects the foraging efficiency and the adaptive ability to deal with the traffic congestion.

4. Results

4.1. Simulation Experiment

We conducted simulation experiments under similar conditions to compare the DRTM with ARTM. Each experiment ran for 20*min* (*T*=1200s) and the simulation algorithm ran 10 times each second, so each experiment involved 12000 simulation steps. The experimental area was a square area of *L*=4.0*m* and the diameter of the nest was *d*=1.5*m*. The total number of robots was N_r=20. The total number of food tokens in the experimental environment was N_f=20, and each food token contained 1500 energy units. Each foraging robot consumed one energy unit per step and each OAE consumed E_a=10 energy units. The food reserve in the nest was consumed at the rate of 2.0 (food tokens)/*min*. Following the ARTM, the other parameter settings were T_1=50 steps, θ_{t0}=3.3 (1 ≤ θ ≤ 10), integer *n* generated randomly (2 ≤ *n* ≤ 9), and *F*(0)=5. To allocate many more robots to go foraging, we set the desired number of food tokens in the nest as F_d=20. However, there were no reference standards for selecting parameters *α* and *β*, so we chose those values by trial and error; in each the experiment, we set *α*=3 and *β*=2. For each TA model, we repeated the simulation experiments 50 times, averaged the results, and then processed the data using MATLAB.

At the beginning of each experiment, there were 5 food tokens in the nest, with the desired number being 20 and the robots are all in the nest. Therefore, the stimulus *S*(*t*) is 15 and threshold θ is 3.3. According to Equation (7), the foraging probability P_s is affected mainly by the stimulus *S*(*t*) and many robots are allocated to foraging. The movements of the active foraging robots consume much more energy, meaning that the number of food tokens in the nest decreases rapidly. As the foraging robots delivery food tokens back to the nest continuously, the number of food tokens in the nest increases rapidly. At this stage, many robots are foraging and the threshold θ becomes larger in the DRTM. As the stimulus *S*(*t*) decreases gradually, the threshold θ and stimulus *S*(*t*) begin together to affect the foraging behavior of the robots. The foraging probability decreases gradually and the rate at which food accumulates decreases. When the number of food tokens within the nest reaches the desired level, the robots stop going foraging. However, because the swarm's nest consumes energy, when the number of food tokens drops below the desired level, the robots begin foraging again.

Figure 5 shows how the amount of food in the nest changes, from which we can calculate the ADF (V_f) and the energy efficiency (E_e). Compared with the ARTM (V_f=4.74), the DRTM (V_f=2.55) maintains a net amount of food preserved in the nest that is closer to the desired level. The ADF (V_f) shows that the DRTM can respond to changes in food supply more effectively than can the ARTM. At the end of the experiment, the total number of food tokens collected by the DRTM is 173.76 and the net number of food tokens in the nest is 20.84; the corresponding values for the ARTM are 183.46 and 18.34, respectively. Each food token contains 1500 energy units. According to Eq. (9), the DRTM (E_e=11.99%) consumes less energy than does the ARTM (E_e=9.99%). The energy efficiency (E_e) shows that the DRTM gives rise to more-efficient foraging.
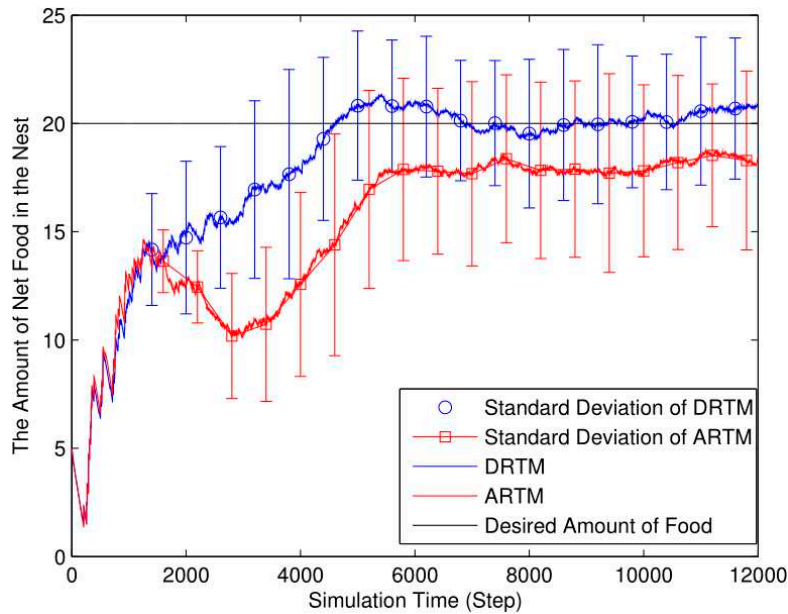
Figure 5. Net number of food tokens in nest (N_f=20, E_a=10, N_r=20, L=4).

As shown in Figure 6, the ARTM generates more OAEs in total (M_T=1148.44) compared to the DRTM (M_T=925.96). In the DRTM, the swarm adjusts the threshold θ in real time according to the traffic conditions, thereby influencing the foraging probability of those robots within state S_W. The DRTM encounters fewer obstacles (M_v=46.30) compared to the ARTM by considering the traffic conditions, and the DRTM possesses better adaptive abilities compared to the ARTM (M_v=57.42) in dealing with traffic congestion.

Figure 6. Number of obstacle avoidance events (OAEs) (N_f=20, E_a=10, N_r=20, L=4).

Table 2 lists the results of t-tests between the DRTM and ARTM regarding the net number of food tokens (F_t) and the total number of OAEs (M_T) (listed are the t value, the two-tailed P value, and the significance of the results). In Table 2, the sample size and number of degrees of freedom were set as 50 and 98, respectively. Based on the results of statistical tests in Table 2, it is clear that the results of DRTM are significantly better than the results of ARTM.

11

Table 2. Results of t-tests between DRTM and ARTM regarding total amount of food (F_t) and total amount of obstacle avoidance (M_T).

Compare	t-value	Two-tailed P	Significance
F_t	3.1918	0.0019	YES
M_T	2.2899	0.0242	YES

4.2. Experiments with Actual Robots

In order to verify the effectiveness of the self-organized task allocation method based on DRTM, the E-puck robot was used to perform the foraging task of swarm robots. The e-puck robots are open tools, which have been used extensively for swarm experiments. With a diameter of 6.8 cm and a height 5.3 cm, each e-puck robot is equipped with (i) eight infrared proximity sensors for detecting obstacles, (ii) one CMOS camera to look for objects, (iii) one three-dimensional accelerometer, (iv) three microphones, and(v)one loudspeaker. Each E-puck robot is equipped with a 3D printed component, as shown in Figure 7(a), which can realize the passive grasping of food and transport the food back to the nest. The food to be searched by the robot is cylindrical, with a diameter of 3.5cm and a height of 4.2cm. As shown in Figure 7(b), the upper half of the food is red and the lower half is white. There is a color camera on the front of the robot with a resolution of 640*480. When the robot searches in the environment, it judges whether it has searched for food by observing the color of the object. The environment was designed as a rectangular area of 1.2m×1.5m, and each robot moved at a speed of v=0.05m/s.
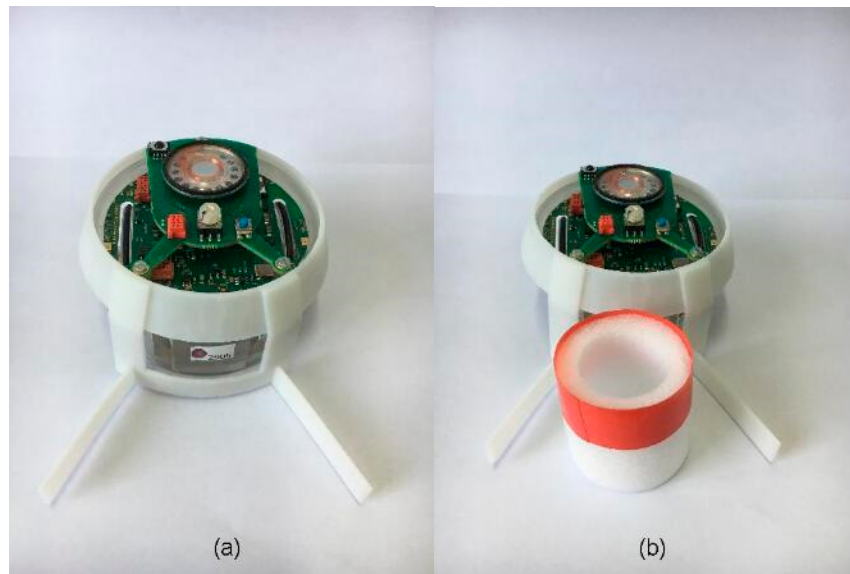

Figure 7. E-puck robot for foraging tasks.

In this experiment, seven E-puck robots were used to complete swarm foraging, as shown in Figure 8. In order to reduce the error and enable the robot to return to the nest effectively, a static robot is placed in the center of the nest as a beacon, and the robot can continuously send out signals to assist the foraging robot to return to the nest. The beacon robot is in the center of the nest, and 6 foraging robots are evenly distributed along the perimeter of the nest. In order to verify the effectiveness of the task allocation method and study the factors affecting the task allocation, swarm-robotics foraging experiments were carried out in two different situations.

Figure 8. Experimental environment for performing swarm foraging.

In the first scenario, the initial food quantity in the nest is set to 10, and the expected food quantity is 10. At the beginning of the experiment, the stimulus value $S(t)$=0, and the robot is in a waiting state in the nest. When part of the food in the nest is consumed by the swarm, the stimulus value $S(t)$ increases, and the robot decides whether to start foraging by calculating the foraging probability. At this time, the TFD and the average number of obstacle avoidance are both 0. From Equation (7), it can be seen that the foraging probability of the robot is mainly affected by the stimulus value $S(t)$. Therefore, as time increases, the robot's foraging probability gradually increases. After the foraging robot successfully forages and returns to the nest, if the amount of food in the nest is greater than the expected amount, the robot stops foraging. During the experiment, the difference between the amount of food in the nest and the expected amount of food is small, so the swarm-robotics system allocates fewer robots for food. According to the experimental observation, when the amount of food in the nest is less than the expected value, the swarm-robotics system always assigns 1-3 robots to look for food. The entire experiment ran for 10 minutes, and the swarm robot achieved autonomous task allocation and maintained the amount of food in the nest at the desired level.

In the second scenario, the initial food quantity in the nest is 0, and the expected food quantity is 10. At the beginning of the experiment, the stimulus value $S(t)$=10, the TFD and the average number of obstacle avoidance are both 0, that is, the threshold θ=0, and the robot has a higher probability of foraging from Equation 7. The system will allocate more robots to start foraging. It is observed from the experiment that at the initial moment, all six robots start to look for food. Simultaneous foraging of multiple robots increases the speed of food collection, but the TFD in the environment also increases. Secondly, the average number of obstacle avoidance among robots also gradually increases, which makes the threshold θ larger. When the robot forages successfully and returns to the nest, due to the decrease of the stimulus value $S(t)$ and the increase of the threshold θ, the number of robots foraging also gradually decreases. Figure 9 shows the experimental state after the experiment runs for a short period of time. At this time, the four robots are in the foraging state. With the increase of time, the amount of food in the nest gradually increased, while the number of foraging robots gradually decreased. When the amount of food in the nest reaches the desired amount, the swarm-robotics system no longer assigns robots to start looking for food. Afterwards, the foraging behavior of the robot is similar to that in Scenario 1, and a small amount of robots are allocated each time to start foraging. The experiment runs for 15 minutes, and the swarm robots can achieve autonomous task allocation and the amount of food in the nest can be maintained at a stable level. The experimental results prove that the task allocation method based on DRTM can achieve effective task allocation and ensure that the swarm robots can complete the foraging task.

Figure 9. Experimental process of swarm foraging.

5. Discussion

The simulation results show that the DRTM gives rise to far-more-efficient foraging compared to the ARTM, but some shortcomings remain with the simulation experiments. In swarm foraging, many factors affect the foraging performance, such as the number of food tokens in the foraging arena (N_f), the energy consumption per OAE (E_a), the total number of robots (N_r), and the size of the foraging arena ($L \times L$). Therefore, it is necessary to explore the swarm foraging further under different experimental conditions, and we take the experiments reported in Sect. 4.1 for comparison. For each TA model, we ran the simulation experiments 50 times.

5.1 Different Numbers of Food Tokens

To study how the number of food tokens (N_f) in the foraging arena affects the foraging performance, we ran experiments for a swarm with $N_f = 25$ and the other parameters the same as in the comparison experiments.

As can be seen in Figure 10, both the DRTM (V_f=2.90) and the ARTM (V_f=2.92) maintain the number of food tokens near the desired level. The total number of food tokens collected by the DRTM is 156.90 and the net food is 20.88; the corresponding values for the ARTM are 150.44 and 16.63, respectively. The DRTM (E_e=13.31%) consumes less energy compared to the ARTM (E_e=11.06%), meaning that the DRTM gives rise to more-efficient foraging.
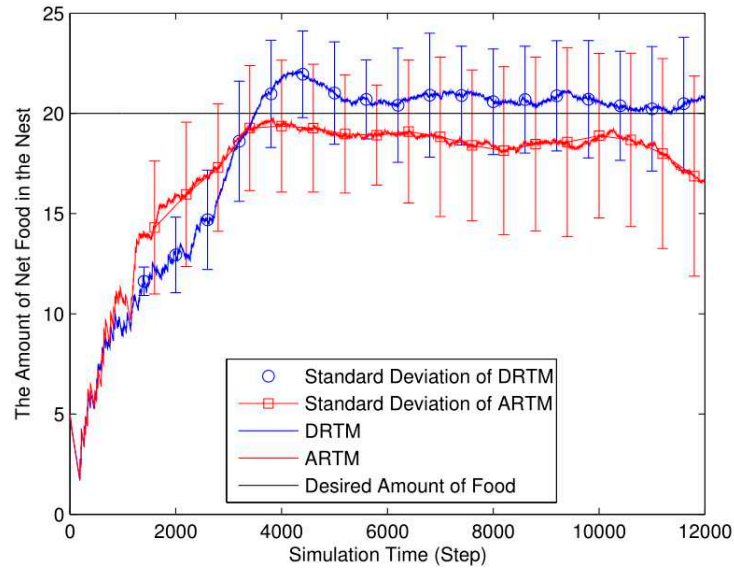
Figure 10. Net number of food tokens in nest (N_f=25, E_a=10, N_r=20, L=4).

In Figure 11, the total number of OAEs for the ARTM (M_T=1053.20) is higher than that for the DRTM (M_T= 760.22). Compared with the ARTM (M_V=52.66), the DRTM (M_V=38.01) has fewer OAEs and possesses stronger adaptive abilities to deal with traffic congestion.
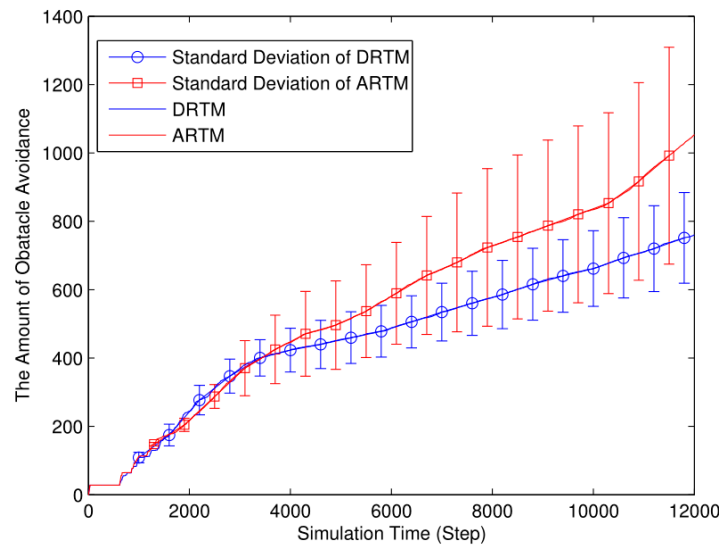

Figure 11. Number of OAEs (Nf = 25, Ea = 10, Nr = 20, L = 4). (N_f=25, E_a=10, N_r=20, L=4).

When there are more food tokens within the foraging arena, the active foraging robots discover them more easily and spend less time foraging. Compared with the comparison experiments, there are fewer OAEs and the energy efficiency increases for both the DRTM and ARTM. Therefore, increasing the amount of food improves the foraging efficiency.

5.2 Different Energy Consumption Due to Obstacle Avoidance

To explore the influence of the energy consumption per OAE (E_a) on foraging efficiency, we ran experiments with E_a=20 and the other parameters the same as in the comparison experiments.

As shown in Figure 12, the DRTM (V_f=3.06) maintains a number of food tokens that is closer to the desired value compared to the ARTM (V_f=5.18). The total number of food tokens collected by the swarm using the DRTM is 189.14 and the net food is 20.39; the corresponding values for the ARTM are 189.32 and 15.99, respectively. According to Eq. (9), the DRTM (E_e=10.78%) consumes less energy than does the ARTM (E_e=8.45%), so the DRTM gives rise to more-efficient foraging.
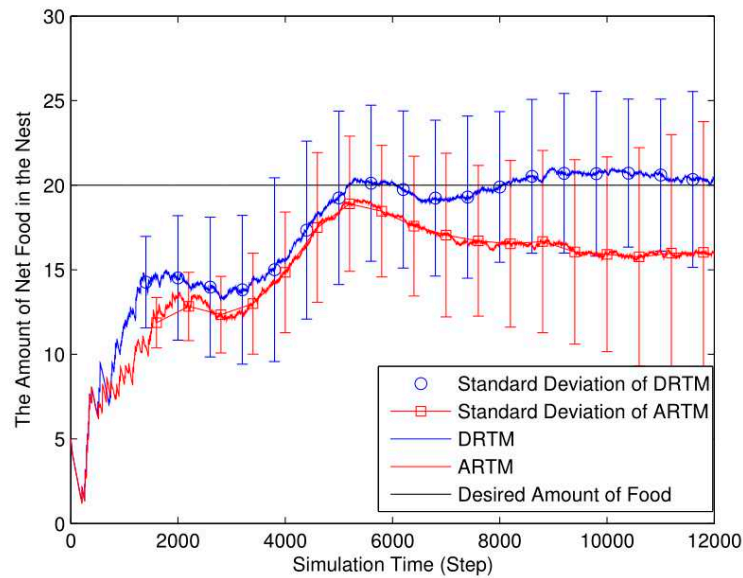
Figure 12. Net number of food tokens in nest (N_f=20, E_a=20, N_r=20, L=4).

Figure 13 plots the number of OAEs for both the ARTM (M_I= 1152.26) and DRTM (M_I=912.06). It is obvious that the average number of OAEs is less for the DRTM (M_V=45.60) compared to the ARTM (M_V=57.61). We conclude that the DRTM copes effectively with changing traffic conditions.
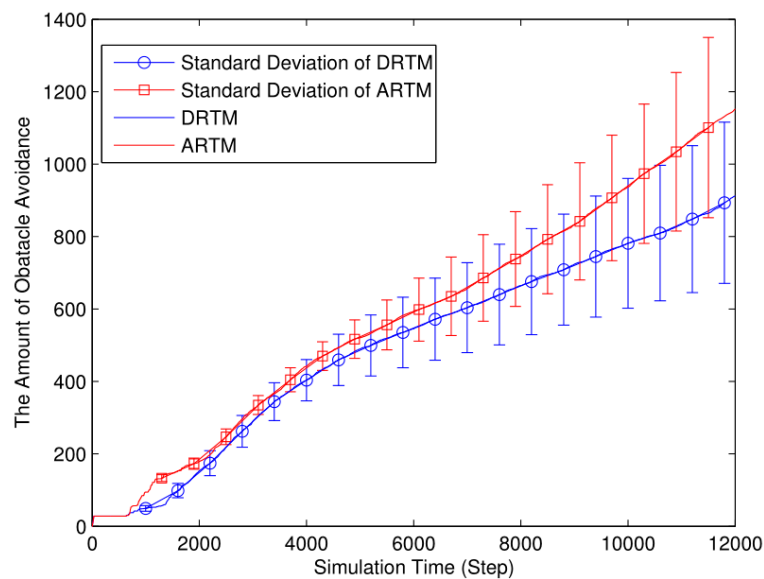

Figure 13. Number of OAEs (N_f=20, E_a=20, N_r=20, L=4).

Increasing the energy consumption per OAE (E_a) decreases the energy efficiency and increases the ADF. To maintain the number of food tokens at the desired level, the swarm allocates many more robots to foraging. This produces many more OAEs, reducing the foraging efficiency further. Therefore, compared with the comparison experiments, increasing the energy consumption per OAE decreases the foraging efficiency.

5.3 Different Numbers of Robots

In swarm foraging, the total number of robots plays an important role in determining the foraging performance. To test how the number of robots affects the swarm foraging, we ran experiments for a swarm with N_r=30 and the other parameters the same as in the comparison experiments.

As can be seen in Figure 14, the DRTM (V_f=3.17) maintains a number of food tokens that is closer to the desired level compared to the ARTM (V_f=4.94). The total number of food tokens collected by the DRTM is 229.44 and the net food is 19.99; the corresponding values for the ARTM are 236.40 and 15.77, respectively. The ARTM (E_e=6.67%) consumes much more energy than does the DRTM (E_e=8.72%), and so the DRTM gives rise to more-efficient foraging.
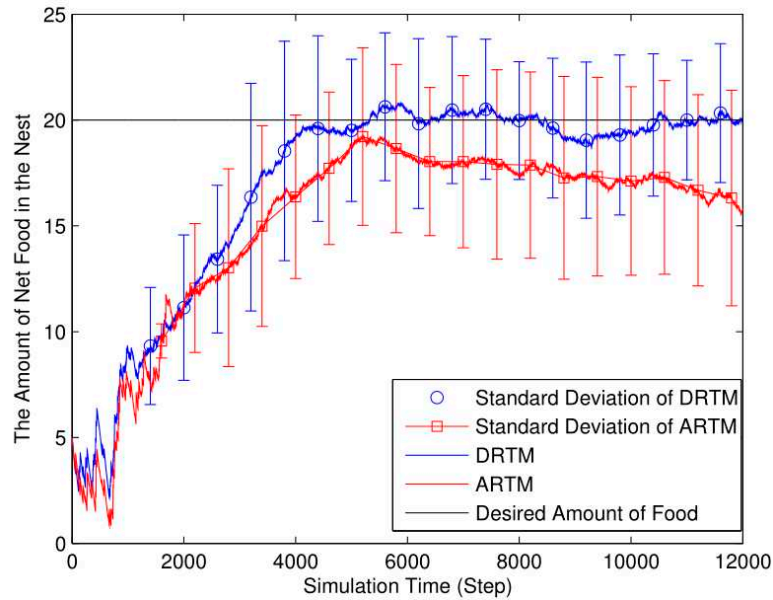

Figure 14. Net number of food tokens in nest (N_f=20, E_a=10, N_r=30, L=4).

In Figure 15, the total number of OAEs for the ARTM (M_T= 1612.26) is clearly higher than it is for the DRTM (M_T= 1203.32). Compared with the ARTM (M_V=53.74), the DRTM (M_V=40.11) has a greater advantage regarding traffic congestion.
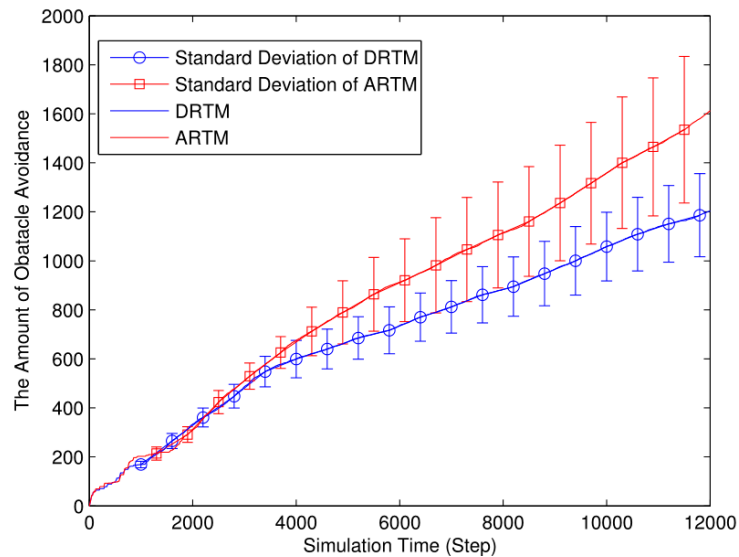

Figure 15. Number of OAEs (N_f=20, E_a=10, N_r=30, L=4).

On one hand, the net number of food tokens approaching the desired level spends less time than in the comparison experiments because many robots forage simultaneously. On the other hand, too many robots foraging simultaneously not only produces many more OAEs but also decreases the probability of finding food tokens because they are limited in number. As can be seen from the experimental data, the energy efficiency, the ADF, and the average number of OAEs all deteriorated. Therefore, increasing the number of foraging robots increases the swarm performance up to a point,

after which the performance starts to decrease influenced by the negative effects of interference among the robots.

5.4 Different Sizes of Foraging Arena

We designed a set of experiments to investigate the effect of the size of the foraging arena on the foraging performance. We ran experiments for a swarm with L=3, d=1, and the other parameters the same as in the comparison experiments.

As shown in Figure 16, the DRTM (V_f=2.35) maintains a number of food tokens that is closer to the desired level compared to the ARTM (V_f=4.61). The total number of food tokens collected by the DRTM is 165.32 and the net food is 17.75; the corresponding values for the ARTM are 170.90 and 15.22, respectively. The DRTM (E_e=10.74%) consumes less energy than does the ARTM (E_e=8.91%), meaning that the DRTM gives rise to more-efficient foraging.
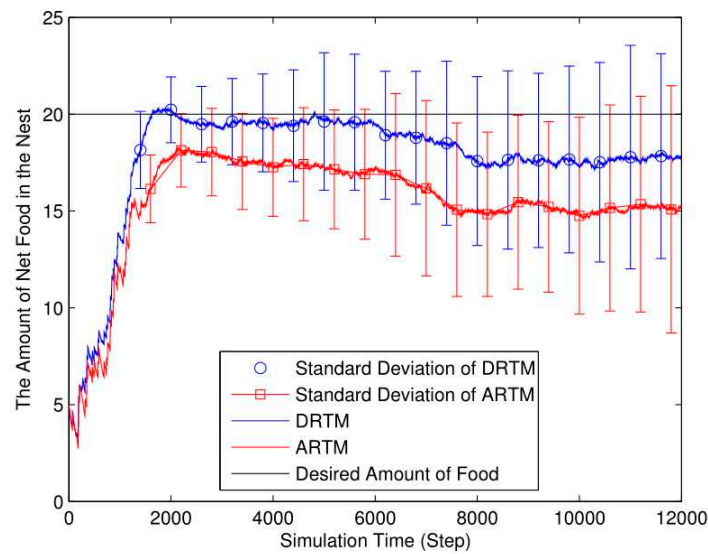

Figure 16. Net number of food tokens in nest (N_f=20, E_a=10, N_r=20, L=3).

As shown in Figure 17, the ARTM swarm (M_T=1882.32) clearly produces many more OAEs than does the DRTM one (M_T=1423.08). By taking the traffic information into account, the DRTM (M_V=71.15) allocates an appropriate number of robots to go foraging and therefore responds more effectively to traffic congestion compared to the ARTM (M_V=94.12).
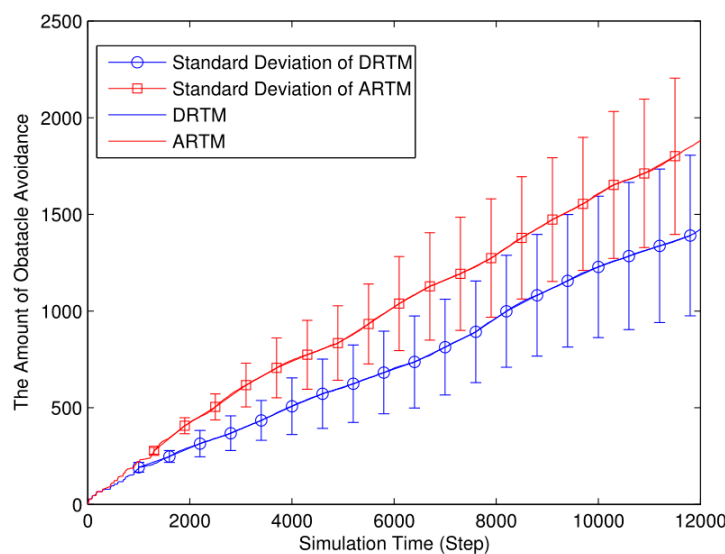

Figure 17. Number of OAEs (N_f=20, E_a=10, N_r=20, L=3).

Foraging robots are more likely to find food in a smaller arena, thereby decreasing the ADF. However, swarm foraging in a smaller arena produces many more OAEs, thereby reducing the energy efficiency. Compared with the comparison experiments, the ARTM swarm produces too many OAEs, and so we conclude that the DRTM deals better with traffic congestion than does the ARTM. Therefore, using too small an arena reduces the foraging efficiency.

Table 3 lists the results of *t*-tests between the DRTM and ARTM for each of the preceding subsections (listed are the t value, the two-tailed P value, and the significance of the results). The sample size and number of degrees of freedom were set as 50 and 98, respectively, for all cases in Table 3. Based on the results of these statistical tests, in Table 3, it is clear that the results of DRTM are significantly better than the results of ARTM.

Table 3. Results of *t*-tests between DRTM and ARTM regarding total amount of food (F_t) and total amount of obstacle avoidance (M_T) for all cases.

Section	Compare	*t*-value	Two-tailed *P*	Significance
5.1	F_t	4.9663	2.89×10^{-6}	YES
	M_T	2.7761	6.59×10^{-3}	YES
5.2	F_t	3.2744	1.46×10^{-3}	YES
	M_T	2.3803	0.0192	YES
5.3	F_t	4.7869	6.00×10^{-6}	YES
	M_T	3.9267	1.60×10^{-4}	YES
5.4	F_t	2.1899	0.0310	YES
	M_T	2.6984	8.21×10^{-3}	YES

In summary, the foraging efficiency was affected by many factors, such as the number of food tokens in the foraging arena (N_f), the energy consumption per OAE (E_a), the total number of robots (N_r), and the size of the foraging arena ($L \times L$). We performed the simulation experiments many times under different conditions, and the experimental results showed that the DRTM gives rise to more-efficient foraging compared to the ARTM.

5. Conclusion

In this paper, we proposed the DRTM that can be used to adjust the number of active foraging robots in a self-organized manner. In the DRTM, the response threshold θ is influenced by the number of active foraging robots and the number of OAEs. When there is traffic congestion in the foraging arena, fewer robots are allocated to foraging. Conversely, when the traffic is flowing smoothly, many more robots tend to go foraging. The external stimulus $S(t)$ is used to measure the difference between the existing number of food tokens and the desired number. An individual robot uses the threshold θ and stimulus $S(t)$ to calculate its foraging probability, based on which the robot determines whether or not to forage. The foraging behavior of those robots in state SW are controlled by the ASM. Finally, simulation experiments were conducted to assess the effectiveness of the DRTM. The experiments verified that the proposed DRTM improves the TA performance and gives rise to more-efficient foraging.

Author Contributions: Data curation, Qingyang Xu; Methodology, Bao Pang and Yong Song*; Software, Ziqi Zhang; Validation, Xianfeng Yuan; Writing – original draft, Bao Pang.

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

References

1. M. Kreiger, J. Billeter. The call of duty: Self-organized task allocation in a population of up to twelve mobile robots. Robotics Autonomous Systems, 2000, 30(1), 65–84.
2. M. U. Arif，S. Haider. A Flexible Framework for Diverse Multi-Robot Task Allocation Scenarios Including Multi-Tasking. ACM Transactions on Autonomous and Adaptive Systems, 2022, 16(1), 1-23.
3. B. Jin, Y. Liang, Z. Han, et. al. A hierarchical training method of generating collective foraging behavior for a robotic swarm. Artificial Life and Robotics, 2022, 27, 137–141.
4. E. Castello, T. Yamamoto, Y. Nakamura, et al. Task allocation for a robotic swarm based on an adaptive response threshold model. 2013 13th International Conference on Control, Automation and Systems (ICCAS). Korea (South), 2013, 259–266.
5. Q. Lu, A. D. Griego, G. M. Fricke, et al. Comparing Physical and Simulated Performance of a Deterministic and a Bio-inspired Stochastic Foraging Strategy for Robot Swarms. 2019 IEEE International Conference on Robotics and Automation (ICRA), Canada, 2019, 9285-9291.
6. K. Di, Y. Zhou, F. Yan, et al. A Foraging Strategy with Risk Response for Individual Robots in Adversarial Environments. ACM Transactions on Intelligent Systems and Technology, 2022, 13(5), 83.
7. R. Miletitch, A. Reina, M. Dorigo, et. al. Emergent naming conventions in a foraging robot swarm. Swarm Intelligence, 2022, 16, 211–232.
8. Y. Khaluf, M. Birattari, F. Rammig. Analysis of long-term swarm performance based on short-term experiments. Soft Computing, 2016, 20(1), 37–48.
9. E. Castello, T. Yamamoto, Y. Nakamura, et al. Foraging optimization in swarm robotic systems based on an adaptive response threshold model. Advanced Robotics, 2014, 28(20), 1343–1356.
10. T. H. Labella, M. Dorigo, J. L. Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2006, 1(1), 4–25.
11. E. Castello, T. Yamamoto, F. D. Libera, et al. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. Swarm Intelligence, 2016, 10(1), 1–31.
12. S. Shaw, E. Wenzel, A. Walker, et. al. ForMIC: Foraging via Multiagent RL With Implicit Communication. IEEE Robotics and Automation Letters, 2022, 7(2), 4877-4884.
13. D. Lee, Q. Lu, T. C. Au. Multiple-Place Swarm Foraging with Dynamic Robot Chains. 2021 IEEE International Conference on Robotics and Automation (ICRA), China, 2021, 11337-11342.
14. E. Ordaz-Rivas, A. Rodriguez-Linan, L. Torres-Trevino. Autonomous foraging with a pack of robots based on repulsion, attraction and influence. Autonomous Robots, 2021, 45, 919–935.
15. M. S. Talamali, T. Bose, M. Haire, et. al. Sophisticated collective foraging with minimalist agents: a swarm robotics test. Swarm Intelligence, 2020, 14, 25–56.
16. S. Adams, D. J. Ornia, M. Mazo. A self-guided approach for navigation in a minimalistic foraging robotic swarm. Autonomous Robots, 2023.
17. D. Lee, Q. Lu and T. C. Au. Dynamic Robot Chain Networks for Swarm Foraging. 2022 International Conference on Robotics and Automation (ICRA), USA, 2022, 4965-4971.
18. K. Di, Y. Zhou, J. Jiang，et. al. Risk-aware collection strategies for multirobot foraging in hazardous environments. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2022, 16(3-4) 1-38.
19. S. O. Obute, P. Kilby, M. R. Dogar, et. al. Swarm Foraging Under Communication and Vision Uncertainties, IEEE Transactions on Automation Science and Engineering, 2022, 19(3), 1446-1457.
20. J. C. Loftus, A. Perez, A. Sih. Task syndromes: linking personality and task allocation in social animal groups. 2021, 32(1), 1-17.
21. A. Swain, S. D. Williams, L. J. Felice, et. al. Interactions and information: exploring task allocation in ant colonies using network analysis. 2022, 189, 69-81.
22. E. Bonabeau, G. Theraulaz, J. L. Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. Bulletin of Mathematical Biology, 1998, 60(4), 753–807.
23. W. Agassounon, A. Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. International Joint Conference on Autonomous Agents and Multiagent Systems, 2002, 1090–1097.
24. W. Liu, A. F. Winfield. Modeling and optimization of adaptive foraging in swarm robotic systems. International Journal of Robotics Research, 2010, 29(14), 1743–1760.
25. E. Buchanan, K. Alden, A. Pomfret, et. al. A study of error diversity in robotic swarms for task partitioning in foraging tasks.Frontiers in Robotics and AI, 2023, 9: 904341.

26.   A. Brutschy, G.Pini, C.Pinciroli, et al. Self-organized task allocation to sequentially interdependent tasks in swarm robotics. Autonomous Agents and Multi-Agent Systems, 2014, 28(1), 101–125.

27.   L. Pitonakova, R. Crowder, S. Bullock. Task allocation in foraging robot swarms: The role of information sharing. in Proceedings of the Fifteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE XV). MIT Press, 2016, 306–313.

28.   E. Bonabeau. Adaptive task allocation inspired by a model of division of labor in social insects. Biocomputing and Emergent Computation: Proceedings of BCEC97, 1997, 36–45.

29.   W. Lee, N. Vaughan, D. Kim. Task Allocation Into a Foraging Task With a Series of Subtasks in Swarm Robotic System. IEEE Access, 2020, 8, 107549-107561.

30.   J. D. Lope, D. Maravall, Y. Quinonez. Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems. Neurocomputing, 2015, 163, 47–55.

31.   A. Kanakia, J. Klingner, N. Correll. A Response Threshold Sigmoid Function Model for Swarm Robot Collaboration. Distributed Autonomous Robotic Systems. Springer, Tokyo, 2016.

32.   A. Kanakia, B. Touri, N. Correll. Modeling multi-robot task allocation with limited information as global game. Swarm Intelligence, 2016, 10(2), 147–160.

33.   T. Yanagida, M. Ueda, T. Murata, et al. Brownian motion, fluctuation and life. Biosystems, 2007, 88(3), 228–242.

34.   A. Kashiwagi, I. Urabe, K. Kaneko, et al. Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. PloS one, 2006, 1(1), e49.

35.   C. F. Daganzo and L. J. Lehe, "Traffic flow on signalized streets," Transportation Research Part B: Methodological, vol. 90, pp. 56–69, 2016.

36.   A. Pradhan, M. Boavida, D. Fontanelli, A Comparative Analysis of Foraging Strategies for Swarm Robotics using ARGoS Simulator. 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Spain, 2020, 30-35.