International Journal of Control, Automation and Systems 17(4) (2019) 1031-1040
http://dx.doi.org/10.1007/s12555-017-0585-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Autonomous Task Allocation in a Swarm of Foraging Robots: An Approach Based on Response Threshold Sigmoid Model

Bao Pang, Yong Song*, Chengjin Zhang, Hongling Wang, and Runtao Yang

Abstract: This paper proposes a task allocation model to adjust the number of working robots autonomously in a swarm of foraging robots. In swarm foraging, the traffic congestion in foraging area and the physical interference between robots can decrease the swarm performance significantly. We introduce the concept of traffic flow density for the first time which can be used to reflect the traffic condition in the foraging area. The amount of obstacle avoidance denotes the number of times physical interference generated in swarm foraging. The traffic flow density and the amount of obstacle avoidance together adjust the value of the threshold. In the proposed response threshold sigmoid model (RTSM), the individual robot can determine autonomously whether to forage or not on the basis of the threshold and the external stimulus and the swarm system can complete the expected foraging task. Simulation experiments are carried out with the aim of evaluating the performance of the proposed method. Several performance measures are introduced to analyze the experimental results and compare to adaptive response threshold model (ARTM). Experimental results verify that the RTSM improves foraging efficiency and decreases the physical interference.

Keywords: Foraging, physical interference, self-organized, swarm robotics, task allocation.

1. INTRODUCTION

In nature, there are many examples where the individual with simple structure, limited function can complete the complicated tasks by means of cooperation with each other [1]. A typical example is the ant colonies foraging, that the ants search for food in the environment and transport the collected food to the nest. No global information and no direct communication, the ants complete the foraging task by local sensing and indirect communication [2]. In ant colonies, the individuals perform a variety of tasks, such as foraging, brood care and nest construction. According to different requirement of the colony, task allocation is used to allocate different number of ants to perform different tasks [3].

Inspired by the complicated social behaviors in ant colonies, this paper employs simple robots to carry out the complex foraging task. The robot in swarm foraging is designed based on swarm intelligence and possesses the basic characteristic of swarm intelligence: decentralized control, limited sensing, and local communication [4]. In the swarm of foraging robots, the individual robot should search for certain objects called for food which is distributed randomly around the foraging area. Once the food is found, the robot collects the food and delivers it to a unique location referred to as the nest [5]. Swarm robotics is mainly in the stage of theoretical research recently and it has not been widely used in real-world applications. However, swarm foraging has lots of potential applications, such as, toxic-waste cleanup, harvesting, land-mine clearance, search and rescue, and area exploration [6]. In these applications, the swarm system can adjust the number of working robots according to the changing environment by using the proposed RTSM.

In recent years, swarm foraging has received widespread attention [7–11]. Without global information and no direct communication, Moon *et al*. construct a virtual pheromone map which effectively realizes the indirect communication among multi-robots [12]. Garnier *et al*. [13] employ chemicals as pheromone and use virtual

Manuscript received September 24, 2017; revised August 24, 2018; accepted November 20, 2018. Recommended by Associate Editor Shun-ichi Azuma under the direction of Editor Doo Yong Lee. This research is supported by the NSFC (National Nature Science Foundation of China)under grant no. 61573213, 61473174, 61473179, by the Natural Science Foundation of Shandong Province under grant no. ZR2015PF009, ZR2017PF008, by the China Postdoctoral Science Foundation under grant no. 2017M612270, by Shandong Province Science and Technology Development Program under grant no. 2014GGX103038, and Special Technological Program of Transformation of Initiatively Innovative Achievements in Shandong Province under grant no. 2014ZZCX04302.

Bao Pang and Hongling Wang are with the School of Control Science and Engineering, Shandong University, Jinan, China (e-mails: pangbao11@foxmail.com, wanghongling720409@163.com). Yong Song, Chengjin Zhang, and Runtao Yang are with the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai, Weihai, China (e-mails: {songyong, cjzhang}@sdu.edu.cn, runtaosd@163.com).
* Corresponding author.

pheromone to realize the communication between robots and eventually they fulfill the swarm foraging task. However, the swarm system can not adjust the number of foraging robots based on the requirement of the swarm and due to simple structures of the robots, the mentioned methods have certain difficulty to implement.

Inspired by division of labor in insect societies, E. Bonabeau introduced a fixed response threshold model (FRTM) [14]. The FRTM can be used to account for the experimental observations of social insects in their division of labor. In ant colonies, when the amount of food in nest is less than a certain threshold (θ), much more ants begin to go foraging; when the amount of food in the nest is greater than a certain threshold (θ), fewer ants begin to go foraging. Individuals with different threshold (θ) respond differently to external stimulus. When given a lower threshold, the individual will respond to a lower external stimulus and the the individual is more likely to go foraging.

As the FRTM is concise and effective to accomplish task allocation, Yang *et al*. [15] have applied the FRTM to the swarm of foraging robots. In swarm foraging, threshold (θ) is a fixed internal variable which is usually given in advance. When a smaller threshold is given, the swarm system has the strong robustness and has better adaptability to against sudden changes; under the same stimulus, much more robots begin to go foraging which will produce more physical interference and thus reduce the foraging efficiency [16]. The swarm with a bigger threshold can decrease the physical interference, but the foraging efficiency is also affected by a lot [17]. Therefore, it can be seen that the swarm with fixed threshold can not achieve expected foraging efficiency. In order to eliminate the limitations of the fixed threshold and further improve the foraging efficiency, some researches improved the foraging efficiency by adjusting the threshold dynamically [18, 19]. A novel response threshold model was improved where the threshold decreased in a predefined amount when a robot performed the task [20]. Castello and Yamamoto [21] proposed an adaptive response threshold model (ARTM) where the threshold (θ) can be adaptively adjusted according to the external stimulus. However, these methods did not consider the traffic condition and physical interference, which will decrease the foraging efficiency.

As noted in [22], the performance of a swarm system is influenced by the physical interference among the foraging robots. Increasement of the number of foraging robots increases the swarm performance up to some point after which the performance starts to decrease influenced by negative effect of the interferences between robots [23]. Pini [24, 25] proposed the task partitioning method, in which the task of moving an object from the source to the nest is partitioned into several sub-tasks consisting of moving the object for a short distance. Although this method decreased the number of physical interference, the time to complete the task was increased which reduced the foraging efficiency. Vaughan *et al*. [26] extended the bucket brigade foraging algorithm in which each robot's foraging area is adapted in response to interference with other robots. Liu *et al*. [27] present an adaptation mechanism to automatically adjust the number of foraging robots to maximize the net energy income to the swarm. By considering the physical interference, these methods reduce the number of foraging robots. Sometimes, although much more interferences will be produced, because of the task demand, much more robots are required to work together to complete the foraging task.

In order to complete the self-organised task allocation according to the requirement of the task and the changing environment, this paper proposes the response threshold sigmoid model (RTSM) to adjust the robots' foraging behaviors. As RTSM contains an exponential term, the robots can respond to the changes of environment in a timely manner. In RTSM, the threshold (θ) can be dynamically adjusted based on the robots' states and the environment situation, such as traffic condition and physical interference. The proposed method is better than the previous methods in the following points.

Firstly, unlike FRTM and ARTM, the RTSM introduces an exponential term which makes the individual robot responds quickly to the changes in the environment. Secondly, RTSM introduces the concept of traffic flow density (TFD) for the first time, the number of the working robots in unit length, to reflect the traffic situation. Also, RTSM employs physical interference, the amount of obstacle avoidance between robots, to measure the traffic condition. The individual robot dynamically adjusts the threshold (θ) according to the amount of obstacle avoidance and TFD, and then decides whether to go foraging or not based on the foraging probability. Thirdly, stimulus $S(t)$, the information about the food in the nest, can evaluate whether the task has completed or not in real time. Because the stimulus $S(t)$ and threshold (θ) together affect the foraging probability, RTSM can allocate different number of robots to go foraging according to the requirement of the task and the changing environment. RTSM can not only guarantee to complete the tasks but also reduce physical interferences.

2. METHODS

2.1. State transitions of swarm foraging

In the study of the swarm foraging, most of the researches focused on the changes of the swarm energy, which are caused by the foraging behaviors and the consumption of the swarm. Swarm energy is all derived from the food collected by the robots and each food contains a certain amount of energy. However, the foraging behaviors of the robots such as the movements of the robots, the physical interference and the swarm in the nest all entail the swarm energy loss. Net energy is the rest of the swarm energy, that is, the energy acquired from the collected food minus the energy consumed by the robots and the swarm. In order to maintain the net energy (net food) at a desired level, the swarm need to implement the self-organized task allocation. In task allocation, the robot determines when to search for food and when to wait in the nest autonomously. Because the net energy and the traffic condition will affect the foraging behaviors, how to improve the foraging efficiency is an important problem in swarm foraging.

In order to research the swarm foraging, we assume that the swarm system used in this study is a homogeneous system. We present a set of behavioral rules which are used to control and guide individual robots to form efficient and adaptive foraging behaviors. All robots are subject to the same behavior rules in performing the foraging task. The foraging behavior of the robots is illustrated by the finite state mechanism shown in Fig. 1. The states for foraging behaviors are as follows:

Wait: At the initial phase of the foraging task, all the robots wait in the nest. T_1 is the minimum waiting time which is given in advance. If the waiting time t_1 that has been consumed by the robot in the **Wait** state is greater than T_1, the individual robot determines whether to go foraging or not. The process from **Wait** to **Search** is the task allocation which will be introduced in detail in next subsection. Based on the foraging probability P_f, the robot chooses whether to switch from **Wait** state to **Search** state or not.

Search: The robot looks for the food based on random walk method, where the robot goes straight until encountering other robots or the boundary and then the robot turns around randomly. In order to better complete the foraging task, we set a maximum search time T_2. If the robot finds food in the given time T_2, it switches to **Cpature** state. If not, it switches to **Return** state.

Return: We set up a light source in the nest, so the robot returns to the nest by tracing the light. By detecting changes of ground color inside and outside the nest, the individual robot ensures whether it has reached the nest or not. When the robot arrives at the nest, it switches to **Wait** state.

Capture: The robot walks up to the food and grab the food. If the robot loses the food or grabs no food, the robot enters **Search** state. If the robot grabs the food successfully, it enters **Delivery** state.

Delivery: The robot with food goes back to the nest by tracing the light. If the robot loses the food, it enters **Search** state. If the robot arrives at the nest, it enters **Deposit** state.

Deposit: The robot unloads the food and then enters **Wait** state.

As can be seen in Fig. 1, the robot within each state can switch to corresponding state if the perceptual cue is satisfied. In **Search** state, the robot searches for food using the random walk method. When the robot encounters obstacles such as other robots and boundaries, the robot turns to the opposite direction to avoid the obstacles. In **Return** state and **Delivery** state, the robot uses light intensity sensors to measure the light intensity to ensure the direction of the nest.

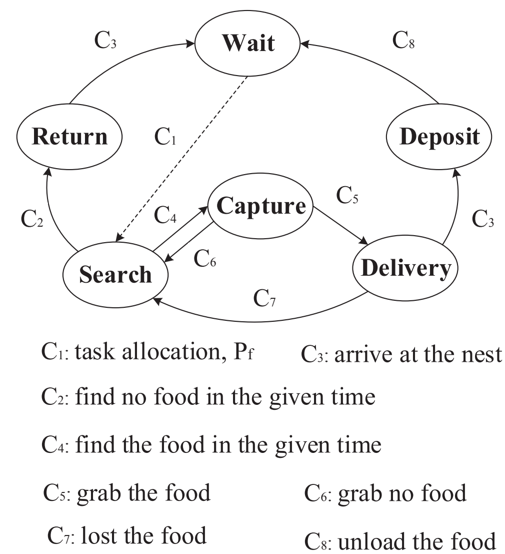


Fig. 1. State transitions of robot's foraging behavior. The foraging behavior can be divided into six states: **Wait**, **Search**, **Return**, **Capture**, **Delivery**, **Deposit**. Each state represents different phases of the foraging task. We define eight perceptual cues ($C_i, i = 1, 2, ..., 8$). Under the influence of perception cues, the individual robot decides whether to switch to the next state. The arrows indicate the relations and directions between the state transitions.

In order to generate self-organized foraging behavior and improve the foraging efficiency, the individual robot needs to know in what circumstances it should go foraging. The perceptual cues C_1 denote the task allocation and the foraging probability P_f determines whether the robot begins to go foraging or not. This paper employs RTSM to calculate the value of P_f and the detailed description is presented in the following subsection.

2.2. Response threshold sigmoid model (RTSM)

The purpose of task allocation is to improve the foraging efficiency and maintain the net food at the desired level. Swarm energy is derived from the food collected by the robots and swarm energy can also be reduced. On the premise of maintaining the net energy at the desired level, how to effectively adjust the robots' behavior is a challenging problem. We employ the RTSM to calculate the foraging probability of each robot. In RTSM, the foraging behavior of individual robot is influenced by the external stimulus ($S(t)$) and the response threshold (θ). The external stimulus $S(t)$ is defined as:

$$S(t) = F_d - F(t), \tag{1}$$

where F_d is the desired amount of food to be maintained in the nest. $F(t)$ refers to the amount of net food at time t in the nest, that is to say, the total food collected by the robots minus the amount of food consumed. Stimulus $S(t)$ denotes the gap between the existing amount of net food and the desired amount of food. When the stimulus $S(t)$ is bigger, the existing amount of net food is much less than the desired level, so much more robots start to go foraging. On the contrary, when the stimulus $S(t)$ is smaller, fewer robots start to go foraging. When $S(t) \le 0$, the robots stop going foraging.

When multiple robots perform foraging task, obstacle avoidance often occurs between the robots. Sometimes obstacle avoidance which occurs among multiple robots can make the robots to turn into the deadlock state. When the robot is in **Return** state or **Delivery** state, the robot goes back to the nest in a straight line by tracing the light; the obstacle avoidance can make the robot deviates from the route, thus the robot spends much more time to return to the nest. Therefore, obstacle avoidance can reduce the foraging efficiency. Obviously, the amount of obstacle avoidance is affected by the number of foraging robots and the size of the foraging area. In order to describe the index quantitatively, we introduce the concept of traffic flow density (TFD) for the first time. TFD is a basic theory in traffic engineering. TFD denotes the number of vehicles within one or several lanes on a unit length of road and it is defined as:

$$k = \frac{N_f}{L}, \tag{2}$$

where N_f denotes the number of vehicles on the road and L is the length of the road; k is the TFD which can be used to measure the density of the vehicles on the road. When k is large, it indicates that there is a traffic congestion on the road. We introduce the TFD to swarm foraging for the first time. In RTSM, N_f denotes the number of active foraging robots and L is the side length of the foraging area. We use (3) to calculate the number of foraging robots.

$$N_f = N_T - N_w, \tag{3}$$

where N_T is the total number of robots in the foraging task and N_w is the number of robots in **Wait** state. Equation (2) can be written as:

$$k = \frac{1}{L} * (N_T - N_w), \tag{4}$$

As can be seen from (4), when the value of k is large, lots of robots are performing the foraging task. Much more obstacle avoidance will be generated and the foraging efficiency will be reduced. Therefore, we should decrease the number of foraging robots. In RTSM, the response threshold changes dynamically and satisfies the following:

$$\theta = \frac{1}{L} * (N_T - N_w) + \frac{1}{N_T} * m, \tag{5}$$

where $\frac{1}{L} * (N_T - N_w)$ is the TFD which can be used to reflect the traffic conditions. In swarm foraging, the amount of obstacle avoidance can also reflect the traffic conditions and it possesses the property of timeliness. Therefore, in (5) the parameter m denotes the amount of obstacle avoidance within a period of time T_3. We initialize $m = 0$ at the beginning of each period of time T_3. N_T is the total number of robots in the foraging task. The larger the value of $\frac{1}{N_T} * m$ is, the larger the influence of the amount of obstacle avoidance on the individual robot is. As can be seen from (5), TFD and the amount of obstacle avoidance together influence the changes of the threshold θ. Therefore, the size of the threshold θ can reflect the traffic condition of foraging area.

In nature, the social insects can complete complex tasks by using self-organized task allocations. The FRTM was proposed to account for the experimental observations of social insects in their division of labor [14]. The FRTM can be denoted as:

$$P_f = \begin{cases} \dfrac{S(t)^n}{S(t)^n + \theta^n}, & S(t) > 0, \\ 0, & S(t) \le 0, \end{cases} \tag{6}$$

where P_f is the state transition probability which determines whether the robots begin to perform foraging task or not. Although the FRTM can realize effective task allocation, sometimes the robots cannot respond to the changing environment timely. In order to overcome this shortcoming, this paper uses the Sigmoid function with the exponential term to make the robots respond quickly to the changes in the environment. The Sigmoid function can be denoted as:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{7}$$

Combining the FRTM and Sigmoid function, this paper proposes the RTSM, where the foraging probability P_f is defined as:

$$P_f = \begin{cases} \dfrac{1}{1 + e^{n(\theta - S(t))}}, & S(t) > 0, \\ 0, & S(t) \le 0. \end{cases} \tag{8}$$

In this equation, the parameter n determines the slope of the probability function. A robot with a larger n is more sensitive to the changes of the environment and can make a rapid response to the stimulus $S(t)$ and threshold

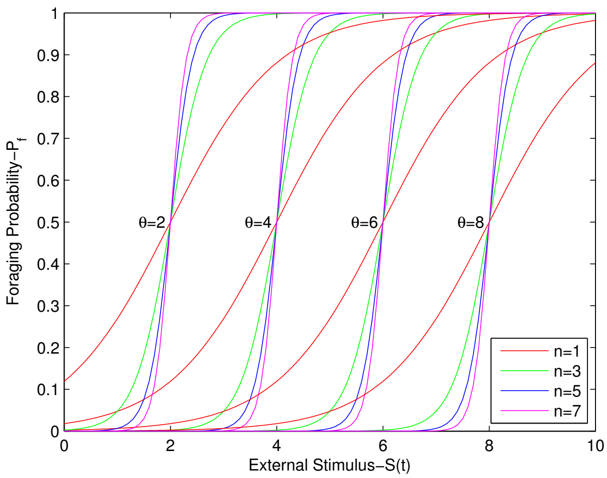


Fig. 2. The probability response curves with different values of θ and n. The larger the n is, the steeper the curve is. When the threshold θ is large, the robots need a bigger stimulus to go foraging.

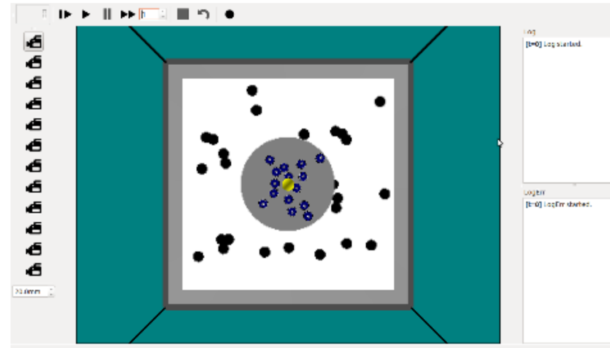


Fig. 3. Screenshot of the simulation experiment in ARGoS simulator. At the beginning of the experiment, all the robots are waiting in the nest.

θ. At the beginning of the foraging experiment, all the robots have the same θ and $S(t)$. If we give the same n to the robots, all the robots will perform foraging task at the same time, which may lead to traffic jams. In order to avoid the traffic jam, we generate the n randomly for each robot. When the amount of net food have reached the desired value, that is $S(t) \leq 0$, all the robots stop performing foraging task and wait in the nest.

As can be seen from Fig. 2, when the stimulus $S(t)$ is large and the threshold θ is small, many robots start to go foraging. On the contrary, when the stimulus $S(t)$ is small and the threshold θ is large, few robots begin to go foraging. When the stimulus $S(t)$ is large and the threshold θ is large, when the stimulus $S(t)$ is small and the threshold θ is small, whether the robot begins to go foraging is determined by the foraging probability P_f. Therefore, whether the robot performs foraging task is influenced by the requirement of the task and the changing environment.

3. EXPERIMENTAL SETUP

3.1. Experimental environment

In this subsection, we set up the simulation experiments to verify the effectiveness of the RTSM. All the simulation experiments are conducted on ARGoS robotic simulator [28]. It is a widely used multi-physics robot simulator and can simulate large-scale swarms of robots efficiently. Fig. 3 is a screenshot of the ARGoS simulator at the beginning of the simulation experiments. As can be seen in Fig. 3, the foraging area is designed as a square area with the side length of L meters. The grey circular region with the diameter in d meter represents the swarm's nest. In the nest, the blue objects are the robots. In the center of the nest, the yellow sphere object is the light source, which can be used to guide directions for robots to trace the light to return to the nest. The black cylindrical objects which are distributed randomly in the foraging area are the food. In the simulation experiment, we use a virtual infrastructure to sense and calculate the total amount of food at the nest and the number of robots in **Wait** state [21]. Once the robot returns to the nest, it sends information about the amount of obstacle avoidance to the infrastructure. After processing and computing, the infrastructure sends these messages to robots that are in **Wait** state. It is important to note that the infrastructure can be seen as the virtual sensor of the swarm and the swarm do not rely on any globalised transmitted communication and any direct communication between robots. Therefore, the RTSM can be regarded as decentralised solutions to foraging processes.

We employ marXbot [29] as experimental robots and Fig. 4 represents a marXbot. The marXbot with a diameter of 0.17 m and height of 0.29 m are placed in the nest at the beginning of the experiment. The robot is equipped with 24 infrared sensors which can be used to perceive obstacles. Eight light intensity sensors are used to measure the intensity of light to find the direction of the nest. Four ground sensors underneath the robot are used to detect the ground color to determine whether the robot has reached the nest or not. A RGB omnidirectional camera is used to discover and identify the food and other robots through different colors. The robots move at speed 0.1 m/s.

To test if the RTSM has the ability to adapt to different environment, we run the experiment with two different areas: $L = 4$ m and $L = 3$ m. Each experiment lasts 1200 s (20 min) and the simulation algorithm runs 10 times each second, so the experiment runs T = 12000 simulation steps. The parameter settings for these experiments are $d = 1.5$ m, $N_T = 15$, the initial amount of food in the nest $F(0) = 5$ and the desired amount of food $F_d = 20$. In order to maintain the net food at the desired level $F_d = 20$, we set a buffer. When the amount of food is more than 21,

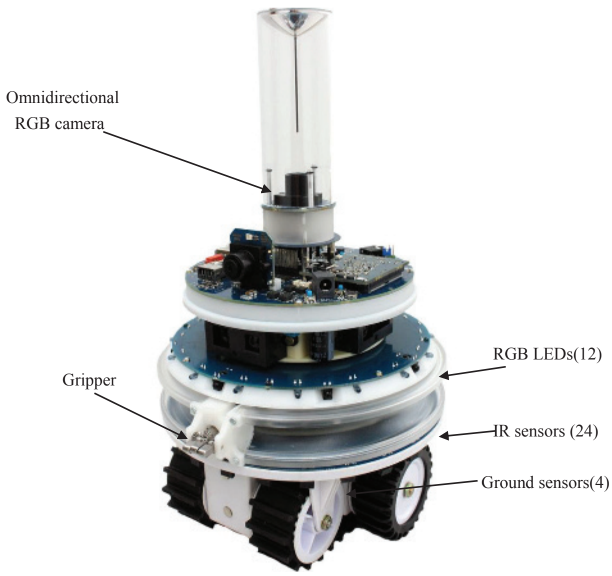


Fig. 4. The marXbot robot. The omnidirectional RGB camera can be used to search for the food. The IR sensors are used to avoid the obstacles and trace the light. The ground sensors can detect the color of the ground to ensure the robot's location.

the robots stop performing the foraging task. When the amount of food is less than 21, the robots begin to perform task allocation. The total amount of food in the environment is 25 and each food contains 1500 energy. The foraging robot consumes 1 energy each simulation step and each obstacle avoidance consumes 10 energy. The consumption rate of the swarm is 5 energy each simulation step.

In RTSM, the parameter settings are $T_1 = 50$ simulation steps, $T_2 = 600$ simulation steps, $T_3 = 300$ simulation steps. In ARTM, the parameter settings are given according to [21]. The threshold parameter θ was initialised to 3.3, $(\theta_{(t_0)} = 3.3)$, $\Delta = 1$, $\alpha = 0.25$. During ARTM execution, θ was bound to the range $(1 \leq \theta \leq 10)$. In all the experiments, this integer n was randomly generated within a predefined range $(2 \leq n \leq 9)$ and remained fixed for the whole experiment.

3.2. Performance measures

We employ three performance indexes to evaluate the effectiveness of the two methods. To maintain the net food at the desired level is one of the purpose of the task allocation in swarm foraging. Therefore, the average deviation of the net food with the desired food is an important index to measure the performance.

The average deviation of food in the nest (V_f) is defined as [21]:

$$V_f = \sum_{t=1}^{T} \frac{|F_d - F(t)|}{T}, \tag{9}$$

where F_d is the desired amount of food that should be maintained in the nest. $F(t)$ is the amount of net food at time t in the nest. V_f denotes the gap between the existing amount of food in the nest and the desired amount of food. The greater value of V_f means that the method can not effectively allocate appropriate number of robots to perform foraging task. Smaller values of V_f imply that the task allocation can maintain food levels closer to the desired value. From this equation, we can see that the value of V_f will be increased regardless of $F(t) > F_d$ or $F(t) < F_d$. Therefore, the average deviation V_f can assess the extent of the task completion and measure the self-organized feature of both methods.

In swarm foraging, most of the researches focused on the changes of swarm energy. As a result, the amount of energy consumed in the foraging task is also considered in the performance index. We employ the energy efficiency (E_e) to assess the methods. E_e is defined as:

$$E_e = \frac{E_n}{E_T}, \tag{10}$$

where E_n is the existing energy, that is to say, the net energy of the remaining food. E_T is the total energy collected by the robots. The greater E_e implies that the robots consume less energy and the corresponding method possesses higher foraging efficiency.

In order to prevent the traffic congestion, this paper takes into consideration of the TFD and the amount of obstacle avoidance which can reflect the traffic condition. In order to verify the effectiveness of this kind of idea, the average amount of obstacle avoidance m_v is also considered as a performance measure:

$$m_v = \frac{m_T}{N_T}, \tag{11}$$

where m_T is the total amount of obstacle avoidance during the mission. N_T is the total number of robots engaged in foraging. m_v denotes the average amount of obstacle avoidance by each robot. Because obstacle avoidance can not only consume energy but also cause traffic jams, m_v can reflect the effect of task allocation intuitively and reflect foraging efficiency to some extent.

4. RESULTS

The simulations of each experimental setup are repeated for 100 runs. RTSM and ARTM all use the response threshold model to calculate the foraging probability in the same simulation environment. In RTSM, (5) is used to obtain the response threshold, where the number of foraging robots and the amount of obstacle avoidance are used. In ARTM, the discrete attractor selection model is used to adapt the response threshold which introduces many more variables. Therefore, the ARTM has a high computation complexity.

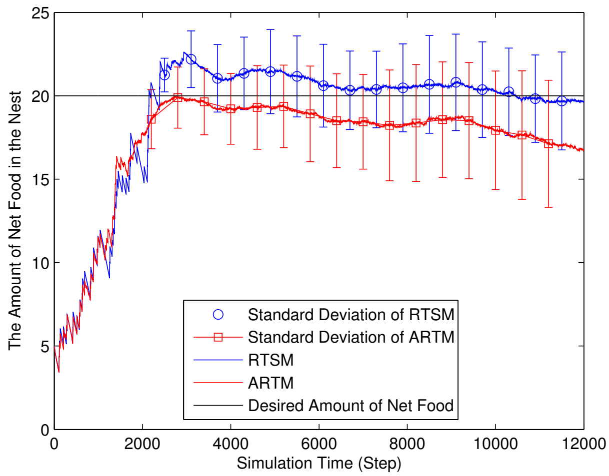


Fig. 5. The amount of net food in the nest during the mission with $L = 4$ m.

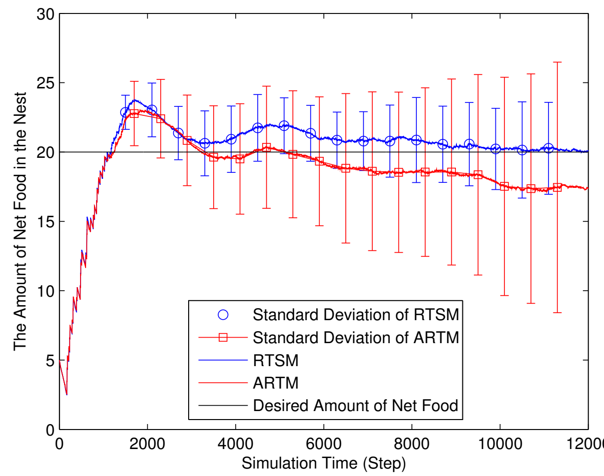


Fig. 6. The amount of net food in the nest during the mission with $L = 3$ m.

The larger foraging areas can reduce the amount of obstacle avoidance between robots, but the individual robot need to search for more areas to find food. Fig. 5 and Fig. 6 depicts the amount of net food in the nest. The amount of net food can be used to calculate the average deviation of food (V_f) and the energy efficiency (E_e). At the beginning of the experiment, there is 5 food in the nest and the desired amount of food is 20. Accordingly, the external stimulus $S(t)$ of each robot is large and the robot's foraging behavior mainly be affected by $S(t)$. Much more robots begin to perform foraging task and the net food in the nest increase rapidly. As the stimulus gradually decreases, the threshold θ and stimulus $S(t)$ begin to together affect the foraging behavior of each robot. The robot's foraging probability decreases gradually and the growth rate of the food slow down. When the amount of net food arrives at the desired value, the robots stop going foraging. The swarm robots in the nest can entail the swarm energy loss and when the amount of food is less than the desired level, the robots start to go foraging again.

As can be seen from Fig. 5, compared with ARTM $(V_f = 2.7583)$, RTSM $(V_f = 2.3022)$ can maintain the food level closer to the desired value. Because the foraging area gets smaller, the robots are more likely to collect food. In Fig. 6, the swarm became faster at collecting food and RTSM $(V_f = 1.6902)$ is better than ARTM $(V_f = 2.0439)$ at keeping the food at the expected level. Moreover, the decrease of the foraging area produces much more obstacle avoidance, which makes ARTM unstable that can be concluded from the standard deviation of ARTM in Fig. 6. The average deviation of food shows that RTSM can make a fast and effective response to the changing environment and possess higher task allocation efficiency.

Each food contains a certain amount of energy, so the energy efficiency can be seen as the food efficiency. In Fig. 5, RTSM $(E_e = 15.09\%)$ consumed less energy than ARTM $(E_e = 13.11\%)$. In Fig. 6, compared with ARTM $(E_e = 15.9\%)$, RTSM $(E_e = 19.3\%)$ improved obviously.

The energy efficiency proves that taking into account the traffic condition, the swarm using RTSM consumes less energy and improves the foraging efficiency effectively.

Fig. 7 and Fig. 8 show how the average amount of obstacle avoidance changes in the experiments, from which the traffic condition and the foraging efficiency can be concluded. At the end of the experiment, the average amount of obstacle avoidance in RTSM $(m_v = 50.5707)$ is significantly less than ARTM $(m_v = 70.7933)$ with $L = 4$ m. When the foraging area was reduced to $L = 3$ m, the average amount of obstacle avoidance in RTSM $(m_v = 50.3093)$ did not change significantly by adjusting the number of foraging robots. However, not considering the traffic condition and the changing environment, the average amount of obstacle avoidance in ARTM $(m_v = 82.8973)$ increased significantly. This result shows that taking into account the traffic condition is an effective method to reduce the amount of obstacle avoidance. RTSM can achieve a better task allocation effect than ARTM in dealing with traffic jam and RTSM possesses higher foraging efficiency.

In swarm foraging, the traffic condition of foraging area can affect the foraging efficiency. To improve foraging efficiency, RTSM takes into account the TFD and the amount of obstacle avoidance to describe the traffic condition quantificationally for the first time. The individual robot determines whether to forage or not based on the traffic condition and the amount of net food in the nest. Accordingly, the swarm robots can decrease the amount of obstacle avoidance under the premise of maintaining

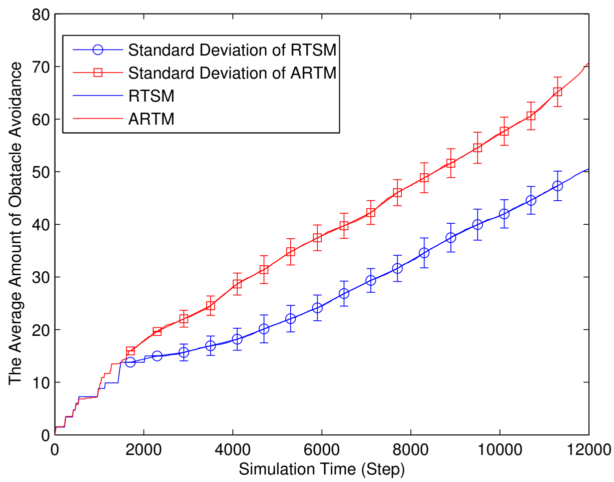


Fig. 7. The average amount of obstacle avoidance with $L = 4$ m. At the beginning of the experiments, the amount of obstacle avoidance grows faster. After that, the growth rate of the RTSM is slower than ARTM.

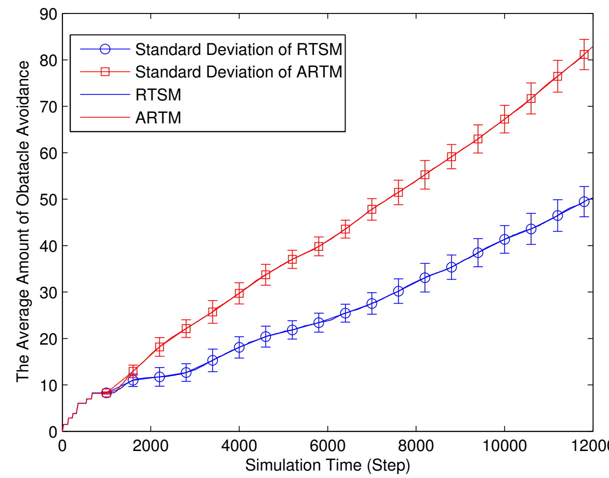


Fig. 8. The average amount of obstacle avoidance with $L = 3$ m.

the food level closer to the desired value. Through the above simulation experiments in different areas ($L = 4$ m, $L = 3$ m), we can see that RTSM showed better performance on all indexes for example the average deviation of food in the nest (V_f), energy efficiency (E_e) and the average amount of obstacle avoidance (m_v) in comparison with ARTM. Therefore, we can conclude that the RTSM improves foraging efficiency effectively. Morever, when foraging area changes, the amount of obstacle avoidance with RTSM almost unchanged. Such experimental result shows that RTSM can adjust the number of foraging robots adaptively according to environmental change, which means that RTSM has robustness for changes in the environment.

5. CONCLUSION AND FUTURE WORK

In swarm foraging, we propose the self-organized task allocation method - RTSM in which the traffic condition is taken into account to improve the foraging efficiency. In RTSM, the response threshold θ is adjusted dynamically according to the traffic flow density and the amount of obstacle avoidance. Using threshold θ and stimulus $S(t)$, individual robot calculates the foraging probability, so as to decide whether to go foraging or not. The robots emerge out of the complex task allocation and foraging behaviors based on the simple rules. Finally, simulation experiments have been given to verify the effectiveness of the RTSM. The experimental results showed that the proposed RTSM can maintain the amount of the net food at the desired level. Moreover, RTSM improves the energy efficiency and reduce the amount of obstacle avoidance, so as to improve the foraging efficiency.

In the proposed method, the food is evenly distributed in the environment and the amount of food is constant. However, the food is usually randomly distributed and the amount of food is changing in practice. Therefore, as future work, we will continue to develop an improved version of RTSM, which can implement the self-organized task allocation under the complex and changeful environment. Morever, time-delays and packet dropouts are frequently encountered in systems, so we will make attempts to extend the current results for the underlying systems under the network-based environment with time-delays, packet dropouts, and quantization [30, 31].

REFERENCES

[1] D. M. Gordon, "The organization of work in social insect colonies," *Nature*, vol. 8, no. 1, pp. 121-124, March 1996.

[2] S. C. Chu, J. F. Roddick, and J. S. Pan, "Ant colony system with communication strategies," *Information Sciences*, vol. 167, no. 1, pp. 63-67, December 2004.

[3] D. M. Gordon, "Interaction patterns and task allocation in ant colonies," *Springer Science & Business Media*, pp. 51-67, 1999.

[4] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, and T. Baaboura, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60-71, September 2013.

[5] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292-321, January 2016.

[6] H. Hamann and H. Rn, "An analytical and spatial model of foraging in a swarm of robots," *Swarm Robotics, Second International Workshop*, pp. 43-55, 2006.

[7] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "A cooperative switching algorithm for multi-agent foraging,"

Engineering Applications of Artificial Intelligence, vol. 50, pp. 302-319, April 2016.

[8] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, "Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution," *Swarm Intelligence*, vol. 11, no. 1, pp. 39-69, February 2017.

[9] L. Pitonakova, R. Crowder, and S. Bullock, "Information flow principles for plasticity in foraging robot swarms," *Swarm Intelligence*, vol. 10, no. 1, pp. 1-31, February 2016.

[10] H. Jung, Y. Kim, and H. K. Dong, "Visual cooperation based on LOS for self-organization of swarm robots," *International Journal of Control, Automation and Systems*, vol. 11, no. 1, pp. 216-224, February 2013.

[11] D. A. Lima and G. M. Oliveira, "A cellular automata ant memory model of foraging in a swarm of robots," *Applied Mathematical Modelling*, vol. 47, pp. 551-572, July 2017.

[12] W. S. Moon, J. W. Jang, H. S. Kim, and K. R. Baek, "Virtual pheromone map building and a utilization method for a multi-purpose swarm robot system," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1446-1453, December 2015.

[13] S. Garnier, F. Tache, M. Combe, and A. Grimal, "Alice in pheromone land: an experimental setup for the study of ant-like robots," *IEEE Swarm Intelligence Symposium*, pp. 37-44, April 2007.

[14] E. Bonabeau, G. Theraulaz, and J. L. Deneubourg, "Fixed response thresholds and the regulation of division of labor in insect societies," *Bulletin of Mathematical Biology*, vol. 60, no. 4, pp. 753-807, July 1998.

[15] Y. Yang, X. Chen, Q. Li, and Y. Tian, "Swarm robots task allocation based on local communication," *International Conference on Computer, Mechatronics, Control and Electronic Engineering*, pp. 415-418, 2010.

[16] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro, "Foraging optimization in swarm robotic systems based on an adaptive response threshold model," *Advanced Robotics*, vol. 28, no. 20, pp. 1343-1356, June 2014.

[17] T. H. Labella, M. Dorigo, and J. L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *Acm Transactions on Autonomous & Adaptive Systems*, vol. 1, no. 1, pp. 4-25, September 2006.

[18] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro, "Task allocation for a robotic swarm based on an adaptive response threshold model," *Proc. of 13th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, Gwangju, pp. 259-266, 2013.

[19] W. Lee and D. Kim, "Adaptive division of labor in multi-robot system with minimum task switching," *Artificial Life 14: International Conference on the Synthesis and Simulation of Living Systems*, pp. 750-756, 2014.

[20] J. D. Lope, D. Maravall, and Y. Qui, "Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems," *Neurocomputing*, vol. 163, pp. 47-55, April 2015.

[21] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach," *Swarm Intelligence*, vol. 10, no. 1, pp. 1-31, January 2016.

[22] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127-141, September 2002.

[23] Y. Khaluf, M. Birattari, and F. Rammig, "Analysis of long-term swarm performance based on short-term experiments," *Soft Computing*, vol. 20, no. 1, pp. 37-48, January 2016.

[24] G. Pini, A. Brutschy, C. Pinciroli, M. Dorigo, and M. Birattari, "Autonomous task partitioning in robot foraging: an approach based on cost estimation," *Adaptive behavior*, vol. 21, no. 2, pp. 118-136, April 2013.

[25] G. Pini, A. Brutschy, A. Scheidler, M. Dorigo, and M. Birattari, "Task partitioning in a robot swarm: object retrieval as a sequence of subtasks with direct object transfer," *Artificial Life*, vol. 20, no. 3, pp. 291-317, August 2014.

[26] A. Lein and R. T. Vaughan, "Adaptive multi-robot bucket brigade foraging," *Artificial Life*, vol. 6, PP. 337–342, January 2008.

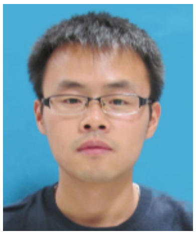
[27] W. Liu, A. F. T. Winfield, J. Sa, J. Chen, and L. Dou, "Towards energy optimization: emergent task allocation in a swarm of foraging robots," *Adaptive behavior*, vol. 15, no. 3, pp. 289-305, September 2007.

[28] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271-295, November 2012.

[29] M. Bonani, V. Longchamp, S. Magnenat, P. Retornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler and F. Mondada, "The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4187-4193, 2010.

[30] J. Qiu, T. Wang, S. Yin, and H. Gao, "Data-based optimal control for networked double-layer industrial processes," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4179-4168, May 2017.

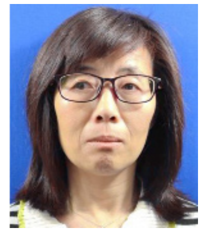
[31] J. Qiu, Y. Wei, H. R. Karimi, and H. Gao, "Reliable control of discrete-time piecewise-affine time-delay systems via output feedback," *IEEE Transactions on Reliability*, vol. 67, no. 1, pp. 79-91, March 2018.



Bao Pang is a Ph.D. candidate in School of Control Science and Engineering at Shandong University. His research interests include swarm robotics, intelligent control, and machine learning.

Yong Song received his Ph.D. degree in Pattern Recognition and Intelligent System from Shandong University in 2012. His research interets include swarm intelligence robotics, machine learning, and neural network.

Chengjin Zhang received his Ph.D. degree in Industrial Automation from Northeastern University in 1997. His research interests include control theory and applications, intelligent robot control, and bioinformatics.



Hongling Wang is a Ph.D. candidate in School of Control Science and Engineering at Shandong University. Her research interests include search and rescue robotics, intelligent robot control, and simultaneous localization and mapping.

Runtao Yang received his Ph.D. degree in Bioinformatics and Intelligent System from Shandong University in 2016. His research interets include swarm intelligence robotics, Bioinformatics, and system biology.