

A MACROSCOPIC PROBABILISTIC MODEL OF ADAPTIVE FORAGING IN SWARM ROBOTICS SYSTEMS

Wenguo Liu, Alan F.T. Winfield and Jin Sa
University of the West of England, Bristol, UK

Corresponding author: Wenguo Liu, Bristol Robotics Laboratory, University of the West of England
Coldharbour Lane, BS16 1QY, Bristol, UK, wenguo.liu@brl.ac.uk

Abstract. In this paper, we extend a macroscopic probabilistic model of a swarm of foraging robots from the homogeneous to the heterogeneous case. In the swarm, each robot is capable of adjusting its searching time and resting time thresholds following the rules described in our previous paper [1]. In order to model the difference between robots, private/public resting time and searching time thresholds are introduced, a number of equations are then developed to work out the relationship between these private time thresholds and public time thresholds based on previously developed difference equations [2]. The extended macroscopic probabilistic model has been tested using the simulation tools Player/Stage. The results from the macroscopic probabilistic model match with those from the simulation with reasonable accuracy, not only in the final net energy of the swarm but also in the instantaneous net energy. Although the model is specific to adaptive foraging, we believe the methodology can be extended to other systems in which the heterogeneity of the system is coupled with its time parameters.

1 Introduction

In swarm robotics, the robots themselves are typically very simple but complex collective behaviours may arise from the interactions among the robots and between the robots and the environment. In order to design and optimise individual robot behaviours and hence achieve the desired collective swarm properties, one of the challenges is to understand the effect of individual parameters on the group performance. Real robot experiments and sensor-based simulations are the most direct way to observe the behaviour of the system with different parameters. However, experiments with real robots, even in simulation, are very costly and time-consuming to implement, and do not scale well as the size of the system grows. It is therefore impractical to scan the whole design parameter space to find the best solutions using a trial and error basis. Mathematical modelling and analysis offer an alternative to experiments and simulations.

In the last few years, more attention has been dedicated to addressing the modelling problem in swarm robotics. Probabilistic models, using both microscopic and macroscopic approaches have been successfully applied to analysis of collective swarm behaviour. A microscopic probabilistic model was first proposed by Martinoli et al. [3, 4] to study collective clustering. The central idea of the microscopic probabilistic model is to describe the interactions among robots and between the robots and environment as a series of stochastic events. The probabilities that each event is triggered are determined by simple geometric considerations and systematic experiments with one or two real robots. Instead of computing the exact trajectories and sensory information of individual robots, as most sensor-based simulations do, state transitions are determined randomly, in effect by throwing dice. Running several series of stochastic events in parallel, one for each robot, allows researchers to study the collective behaviour of the swarm. To obtain statistically significant results, several runs of the model need to be carried out and the overall behaviour of the system is computed by averaging the results of those runs.

Unlike the microscopic model, a macroscopic model directly describes the overall collective behaviour of the system. In general, macroscopic models are more computationally efficient than their microscopic counterparts. One of the fundamental elements of the macroscopic probabilistic model are the Rate Equations, which have been successfully applied to a wide variety of problems in physics, chemistry, biology and the social sciences. For instance, Sumpter and Pratt [5] developed a general framework for modelling social insect foraging systems with generalised rate functions (differential equations). Sugawara and coworkers [6, 7] first presented a simple macroscopic model for foraging in a group of communicating and non-communicating robots, with analysis under different conditions; further study can be found in [8]. Lerman and Galstyan [9, 10] proposed a more generalised and fundamental contribution to macroscopic modelling in multi-agent systems. In [11], they presented a mathematical model of foraging in a homogeneous multi-robot system to understand quantitatively the effects of interference on the performance of the group. In [12], they developed a macroscopic model of collaborative stick-pulling, and the results of the macroscopic model quantitatively agree with both embodied and microscopic simulations. Agassounon and Martinoli [13] use the same approach to capture the dynamics of a robot swarm engaged in collective clustering experiments.

Rather than using a time-continuous model, Martinoli and coworkers [14, 15, 16] considered a more fine-grained macroscopic model of collaborative stick-pulling which takes into account more of the individual robot behaviours, in the discrete time domain using difference equations. They suggested that time-discrete models are the most appropriate solution for the level of description characterised by logical operators and behavioural states. Similarly,

Correll et al. [17, 18] used a macroscopic probabilistic model for analysis of beaconless and beacon-based strategies for a swarm turbine inspection system, and furthermore to find an optimal collaboration policy minimising the time to completion and the overall energy consumption of the swarm in [19, 20]. In [21], a macroscopic probabilistic model is proposed to analyse the self-organised robot aggregation inspired by a study on aggregation in gregarious arthropods.

Despite the success of the above examples, there is very little existing work on mathematical analysis of adaptive multi-robot systems in dynamic environments, with the notable exception of the work done by Lerman and Galstyan [22, 23, 24]. They have extended the macroscopic probabilistic model to study distributed systems composed of adaptive robots that can change their behaviour based on their estimates of the global state of the system. In their study, a group of robots engaged in a puck collecting task need to decide whether to pick up red or green pucks based on observed local information. The heterogeneities in the robot population must therefore be taken into account. They claim that the model can be easily extended to other systems in which robots use a history of local observations of the environment as a basis for making decisions about future actions.

In our previous work [1], we presented a simple adaptation algorithm for robots engaged in a collective foraging task. The adaptation algorithm has a number of parameters which are used to adjust the contribution of each cue. However, with a set of intuitively chosen parameters, it is not clear that the swarm reaches the best performance it can achieve, and there are no obvious guidelines for manually finding the best set of parameters for the algorithms. To address these problems we first developed a macroscopic probabilistic model of collective foraging for a simplified case in [2], where each robot is given the same resting time and searching time thresholds without adaptation. In this paper we will extend the model for the swarm with the adaptation abilities introduced in [1].

2 Collective Foraging with Adaptation

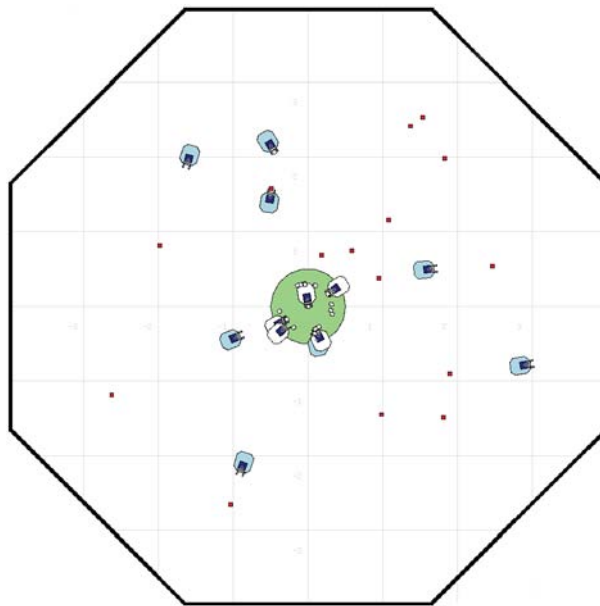


Figure 1: Screen shot of collective foraging in the Player/Stage simulator.

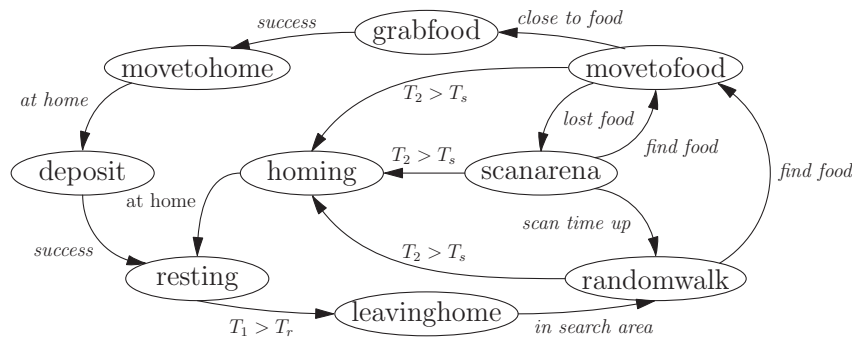


Figure 2: The threshold-based robot controller for collective foraging.

Figure 1 illustrates the collective foraging experiment in a sensor-based simulator – Player/Stage [25]. There are a number of food-items randomly scattered in the arena and as food is collected more will *grow* to replenish the

supply. Each food-item collected will deliver an amount of energy to the swarm but the activity of foraging will consume a certain amount of energy at the same time. A threshold-based controller is implemented to complete the task, as shown in Figure 2. In order to improve the energy efficiency, the individuals in the swarm use three adaptation cues – internal cues (successful or unsuccessful food retrieval), environmental cues (collision with other robots while searching) and social cues (teammate food retrieval success or failure) – to dynamically regulate two internal thresholds, resting time and searching time. Let i ($= 1, 2, \dots, N$) indicate the ID for each robot and T_s^i and T_r^i be the searching time and resting time threshold respectively, then according to [1],

$$T_s^i(k+1) = T_s^i(k) - \alpha_1 C^i(k) + \beta_1 P_s^i(k) - \gamma_1 P_f^i(k) \quad (1)$$

$$T_r^i(k+1) = T_r^i(k) + \alpha_2 C^i(k) - \beta_2 P_s^i(k) + \gamma_2 P_f^i(k) - \eta R^i(k) \quad (2)$$

where $C^i(k)$ counts the collisions while searching, and α_1 and α_2 are adjustment factors to moderate the contribution of the environmental cues. $P_s^i(k)$ and $P_f^i(k)$ represent the social cues from teammates through the stigmergy-like mechanism. The contribution from social cues is moderated by altering the adjustment factors β_1 , β_2 , γ_1 and γ_2 . $R^i(k)$ then donates the internal cues and η is the corresponding adjustment factor. $C^i(k)$, $R^i(k)$, $P_s^i(k)$ and $P_f^i(k)$ in Equation (1) (2) are defined as follows.

$$C^i(k) = \begin{cases} 1 & \text{state } \mathbf{randomwalk} \rightarrow \text{state } \mathbf{avoidance} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$R^i(k) = \begin{cases} 1 & \text{state } \mathbf{deposit} \rightarrow \text{state } \mathbf{resting} \\ -1 & \text{state } \mathbf{homing} \rightarrow \text{state } \mathbf{resting} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$P_s^i(k) = \begin{cases} 0 & \text{not in } \mathbf{resting} \text{ state} \\ SP_s(k) & \text{state } \mathbf{deposit} \rightarrow \text{state } \mathbf{resting} \\ \sum_{i=1}^N \{R^i(k) | R^i(k) > 0\} & \text{in } \mathbf{resting} \text{ state} \end{cases} \quad (5)$$

$$P_f^i(k) = \begin{cases} 0 & \text{not in } \mathbf{resting} \text{ state} \\ SP_f(k) & \text{state } \mathbf{homing} \rightarrow \text{state } \mathbf{resting} \\ \sum_{i=1}^N \{|R^i(k)| | R^i(k) < 0\} & \text{in } \mathbf{resting} \text{ state} \end{cases} \quad (6)$$

where SP_s and SP_f represent the gradual decay rather than instantly disappearing social cues (successful and failure retrieval), which are defined as follows:

$$SP_s(k+1) = SP_s(k) - \delta_s + \sum_{i=1}^N (R^i(k) | R^i(k) > 0) \quad (7)$$

$$SP_f(k+1) = SP_f(k) - \delta_f + \sum_{i=1}^N (|R^i(k)| | R^i(k) < 0) \quad (8)$$

Attenuation factors δ_s and δ_f are introduced here to simulate somewhat akin to ants leaving a decaying pheromone trail while foraging. As the social cues are only accessible for the robots in the nest, two categories of robots will be affected. One group are those already resting in the nest, the other are those ready to move to state *resting* from states *homing* or *deposit*; the former can ‘monitor’ the change of social cues and then adjust its time threshold parameters, while the latter will benefit from the gradually decaying cues deployed by teammates. These two situations for updating $P_s^i(k)$ and $P_f^i(k)$ are shown in Equation (5) (6).

3 Macroscopic Probabilistic Model for a Heterogeneous Swarm

3.1 Probabilistic Finite State Machine (PFSM)

The collective foraging task can be described as a PFSM as shown in Figure 3. Each block in the PFSM represents the corresponding state and the average number of robots in that state, which is marked with N_X . For simplicity, we make some changes from the finite state machine (FSM) shown in Figure 2: the original 9 states are merged into 5 states: states *movetohome* and *Deposit* in the FSM correspond to state *Deposit* (D) in PFSM, states *leavinghome*, *randomwalk* and *scanarena* in the FSM are merged into state *Searching* (S), states *movetofood* and *grabfood* in the FSM are now replaced with state *Grabbing* (G), states *Resting* (R) and *Homing* (H) in the FSM remain the same. The transitions from one state to another are normally based on certain probabilities shown in the edge of the transition lines. For example, γ_f indicates the probability that the robots in state *Searching* will find food and thus transfer to state *Grabbing*. The transitions between two states without probability label shown in the edge are delayed for certain period but with probability 100%. For instance, the transitions from state *AvoidanceS* to *Searching* will be delayed for T_A steps after the robots move to state *AvoidanceS*. The dynamics of the system can

then be captured using a group of difference equations (DE), with each representing the number of robots and its changes in corresponding state. [2] gives more detailed derivations for the DEs and transition probability from first principles.

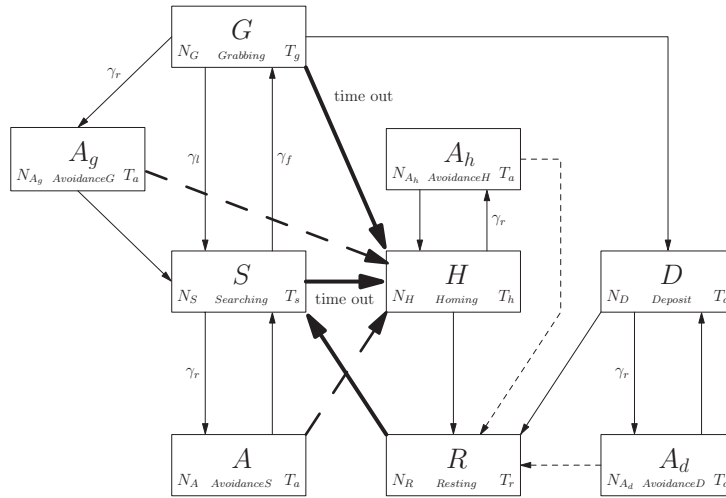


Figure 3: The probabilistic finite state machine for collective foraging. Transitions marked as bolder lines are affected by the variation of time threshold parameters during adaptation.

3.2 The Challenge and its Solution for Modelling Adaptation

Unlike the swarm modelled in [2], here the time threshold parameters (T_s and T_r) for the robots with adaptation abilities will not only vary from one robot to another, but also be different from one time step to the next. However, the basic behaviours for the robots are essentially the same except for some transitions among states which rely on these two time parameters. More specifically, as shown in Figure 3, the transition from state *Searching* or *Grabbing* (including *avoidance*) to state *Homing*, and the transition from state *Resting* to state *Searching* depend on the value of these two time threshold parameters. The changes of T_s and T_r result in the different actions the robots will take. Since the macroscopic model itself doesn't take the difference among individual robots into account, all the parameters and variables used in the model are presented from statistical aspects, for instance, the average number of robots in each state. The challenge is how to introduce these differences into our previously developed macroscopic model. To solve these problems, in conjunction with the sub-PFSM presented in [2], we introduce the concept of private resting time and searching time thresholds, and their counterparts – public resting time and searching time thresholds, into our model. The private time thresholds play the role of deciding when the transition from one state to another is triggered, while the public time thresholds are used to accumulate the contributions from all the adaptation cues which have been applied to the swarm. They affect each other in a bi-directional manner. For example, as shown in Figure 4, the private resting time thresholds in the model are 'inherited' from the public resting time \widehat{T}_r when they are formed (with the robots moving to state *Resting*) and then 'merged' into the public resting time \widehat{T}_r after that subset of robots move to state *Searching*. By defining the 'inherit', 'adaptation' and 'merge' operations according to the adaptation algorithms described above, we can then extend the macroscopic model for a swarm of foraging robots with adaptation abilities and study the effect of these adjustment parameters on the performance of the system. In the following sections, we will deal with the two time thresholds separately.

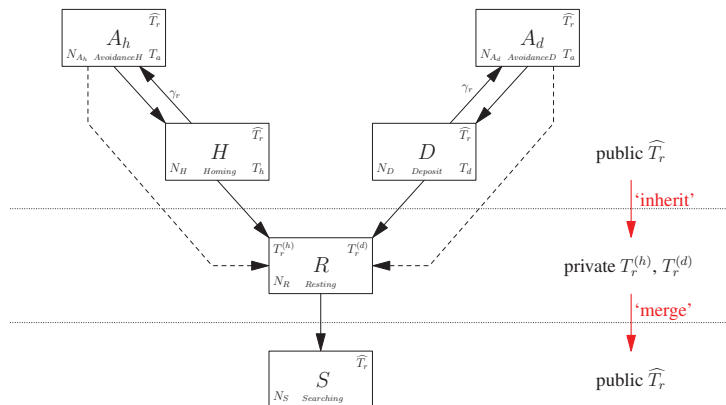


Figure 4: Relationship between private and public resting time threshold parameters. Here 'inherited' means 'making a copy', while 'merge' refers to the update of the \widehat{T}_r based on some rules

3.3 Modelling Adaptation on Resting Time Threshold

The adjustment for each robot falls into three categories corresponding to the contribution from internal cues, social cues and environmental cues respectively. In the model, internal cues and social cues will be applied to adjust the private resting time threshold first, then the changes of private resting time thresholds can be combined together and affect the public resting time threshold \widehat{T}_r afterwards. Meanwhile environmental cues have a direct impact on the adjustment of \widehat{T}_r .

3.3.1 Internal Cues & Social Cues

The effect of internal cues and social cues on resting time thresholds can be categorised into two stages: firstly, the internal cues and social cues are applied when the transitions to state *Resting* occur; then, the social cues continue to play roles on adjustment when the robots are in state *Resting*. In order to model the difference among the individual robots, we need to deal with the ‘failed’ and ‘successful’ robots separately. In our model, each type of robot is endowed with a private resting time threshold $T_r^{(h)}$ or $T_r^{(d)}$ ¹, which is ‘inherited’ from the public resting time threshold \widehat{T}_r . Similar to the sub-PFSMs presented in [2], each $T_r^{(h)}$ and $T_r^{(d)}$ have their own lifetimes. Here notation $T_r^{(h)}(k; i)$ and $T_r^{(d)}(k; i)$ are introduced to represent the values of private resting time threshold, where i indicates the time step that the corresponding private resting time threshold is formed and k denotes the current time step. Taking $T_r^{(d)}(k; i)$ as an example, mathematical modelling of private time thresholds can be summarised to three steps:

- 1) initialising when they are formed ($k = i$);
- 2) adapting during their lifetime ($i < k < i + T_r^{(d)}(k; i)$);
- 3) merging to public time threshold at the end of their lifetime ($k = i + T_r^{(d)}(k; i)$).

Initialisation

As soon as the robots move into state *Resting*, a private resting time threshold is formed for these robots with the initial value of public resting time threshold. The internal cues will be applied to adjust the private resting time threshold first, and then the social cues applied according to the gradually evaporating virtual pheromones deployed by previous returning robots. Following Equation (2), we have

$$T_r^{(h)}(k+1; k+1) = \widehat{T}_r(k) - \beta_2 SP_s(k) + \gamma_2 SP_f(k) + \eta \quad (9)$$

$$T_r^{(d)}(k+1; k+1) = \widehat{T}_r(k) - \beta_2 SP_s(k) + \gamma_2 SP_f(k) - \eta \quad (10)$$

The first term in the right-hand side (RHS) of Equation (9) and Equation (10) represents the ‘inherit’ operation from the public resting time threshold. The second and third terms in the RHS count the contribution of social cues. The last term then depicts the adjustment of internal cues.

As for SP_f and SP_s , the increased value at each time step equals the number of robots returning home (denoted with $\Delta_H(k - T_h)$ and $\Delta_D(k - T_d)$ respectively). Meanwhile, they will ‘evaporate’ with time elapsing. Thus we have

$$SP_f(k+1) = SP_f(k) - \delta_f + \Delta_H(k - T_h) \quad (11)$$

$$SP_s(k+1) = SP_s(k) - \delta_s + \Delta_D(k - T_d) \quad (12)$$

Adaptation

When the robots are already in state *Resting*, i.e. $i < k < i + T_r^{(d)}(k; i)$, they adjust their time thresholds based on the change of social cues, which is equivalent to the increased number of returning robots. The adaptation can be described as

$$T_r^{(h)}(k+1; i) = T_r^{(h)}(k; i) - \beta_2 * \Delta_D(k - T_d) + \gamma_2 * \Delta_H(k - T_h) \quad (13)$$

$$T_r^{(d)}(k+1; i) = T_r^{(d)}(k; i) - \beta_2 * \Delta_D(k - T_d) + \gamma_2 * \Delta_H(k - T_h) \quad (14)$$

where $i < k < i + T_r^{(d)}(k; i)$.

Merging

Once the resting robots move into state *Searching*, the corresponding private resting time thresholds will update the public resting time threshold. The updating of public resting time threshold is referred to as a merging operation. At each time step, there may be more than one group of resting robots running out of their resting time. In order to calculate the contribution that the private resting time thresholds make on the public resting time threshold \widehat{T}_r , we need to know:

¹The superscripts (h) and (d) indicate that which state the robots are transferred from, here (h) for state *Homing* while (d) for state *Deposit*. Similarly, for the private searching time threshold introduced later, (s) represents state *Searching*.

- the number of robots which leave the state *Resting* at the current time step;
- the impact of social cues and internal cues on the private resting time thresholds $T_r^{(h)}$ and $T_r^{(d)}$ during their lifetime.

Since there is only one copy of private resting time threshold $T_r^{(h)}$ formed each time step, the number of resting robots transferring from state *Homing* and their corresponding private resting time thresholds $T_r^{(h)}$ can be identified using the date of birth (DOB). For example, the number of robots transferring from state *Homing* at time step i equals $\Delta_H(i - T_h)$, and their private resting time threshold is $T_r^{(h)}(k; i)$. To calculate the number of resting robots running out of resting time at time step k , we introduce two help-variables, $\mathbb{R}_H(k)$ and $\mathbb{R}_D(k)$, to represent the collection of DOBs of private resting time thresholds which are ready to ‘merge’ into \widehat{T}_r at time step k . then

$$\mathbb{R}_H(k) = \{i | k - i = \lfloor T_r^{(h)}(k; i) \rfloor\} \quad (15)$$

$$\mathbb{R}_D(k) = \{i | k - i = \lfloor T_r^{(d)}(k; i) \rfloor\} \quad (16)$$

Let $\Delta_{S \leftarrow R^{(h)}}(k)$ and $\Delta_{S \leftarrow R^{(d)}}(k)$ be the number of robots transferring from state *Resting* (from state *Homing* and *Deposit* respectively) to *Searching* at time step k , then we have

$$\Delta_{S \leftarrow R^{(h)}}(k+1) = \sum_{i \in \mathbb{R}_H(k)} \Delta_H(i - T_h) \quad (17)$$

$$\Delta_{S \leftarrow R^{(d)}}(k+1) = \sum_{i \in \mathbb{R}_D(k)} \Delta_D(i - T_d) \quad (18)$$

Thus, the total number of robots moving from state *Resting* to *Searching* can be expressed as

$$\Delta_S(k+1) = \Delta_{S \leftarrow R^{(h)}}(k+1) + \Delta_{S \leftarrow R^{(d)}}(k+1) \quad (19)$$

The contribution of each fraction of reactive (from state *Resting* to *Searching*) robots to the public resting time threshold can be expressed as the product of the quantity of the robots and the change of the corresponding private resting time threshold. For instance, assume that a group of resting robots which transferred from state *Homing* become reactive at time step k , if the corresponding private resting threshold has a DOB i , then the contribution from these robots is given by:

$$\Delta_H(i - T_h) \times (T_r^{(h)}(k; i) - \widehat{T}_r(i - 1))$$

where $\widehat{T}_r(i - 1)$ depicts the value of public resting time threshold ‘inherited’ by the $T_r^{(h)}$ when it is formed. Let $\Delta_{T_r^{(h)}}$ and $\Delta_{T_r^{(d)}}$ be the total contribution provided by the resting robots transferred from state *Homing* and *Deposit* respectively at time step k , then

$$\Delta_{T_r^{(h)}}(k) = \sum_{i \in \mathbb{R}_H(k)} \Delta_H(i - T_h) \times (T_r^{(h)}(k; i) - \widehat{T}_r(i - 1)) \quad (20)$$

$$\Delta_{T_r^{(d)}}(k) = \sum_{i \in \mathbb{R}_D(k)} \Delta_D(i - T_d) \times (T_r^{(d)}(k; i) - \widehat{T}_r(i - 1)) \quad (21)$$

The updating of the public resting time threshold for the swarm is then based on the following equation

$$\widehat{T}_r(k+1) = \widehat{T}_r(k) + \frac{\Delta_{T_r^{(h)}}(k) + \Delta_{T_r^{(d)}}(k)}{N_0} \quad (22)$$

where N_0 is the total number of robots in the swarm.

3.3.2 Environmental Cues

The environmental cues play roles in adjusting the resting time threshold for the robots working in the arena (non-resting). Although the change of the resting time threshold in this case will not affect the behaviour of the robots until they return home, the public resting time threshold \widehat{T}_r is changed in the following manner with the environmental cues.

$$\widehat{T}_r(k+1) = \widehat{T}_r(k) + \frac{\alpha_2 * \Delta_A(k+1)}{N_0} \quad (23)$$

where $\Delta_A(k+1)$ depicts the number of robots moving into state *AvoidanceS* at time step k .

3.3.3 Combining all the Cues

Combining the effect of all the cues, the swarm will update its public resting threshold \widehat{T}_r in this manner

$$\widehat{T}_r(k+1) = \widehat{T}_r(k) + \frac{\Delta_{T_r^{(h)}} + \Delta_{T_r^{(d)}} + \alpha_2 * \Delta_A(k+1)}{N_0} \quad (24)$$

3.4 Modelling Adaptation on Searching Time Threshold

Similarly, three private searching time thresholds, $T_s^{(h)}$, $T_s^{(d)}$ and $T_s^{(s)}$, and one public searching time threshold \widehat{T}_s are introduced to model the adaptation of searching time threshold. Figure 5 demonstrates the relationship between the private and public searching time thresholds. Generally, the private searching time thresholds have their own life cycles during the adapting process. They ‘inherit’ the up-to-date \widehat{T}_s when they are formed, and will update (‘merge’) into \widehat{T}_s at the end of their lifetime. However, the situation here is more complex than for the resting time threshold, as the private searching time thresholds occur in different states. Consequently, two pairs of ‘inherit’ and ‘merge’ operations are applied to regulate the exchange of private and public searching time thresholds. Among these three private searching time thresholds, $T_s^{(h)}$ and $T_s^{(d)}$ are used to track the contribution of social cues when the robots are in state *Resting*, while $T_s^{(s)}$ is used to track the contribution of environmental cues. Moreover, as shown in Figure 5, the transition from state *Searching* to *Homing* is now decided by $T_s^{(s)}$. Although $T_s^{(h)}$ and $T_s^{(d)}$ do not change the behaviour of searching robots directly, they have large contributions in adjusting the public searching time threshold \widehat{T}_s , which in return affect the behaviour of searching robots. The update of \widehat{T}_s from these private searching time thresholds can be categorised as two stages according to the social cues and environmental cues.

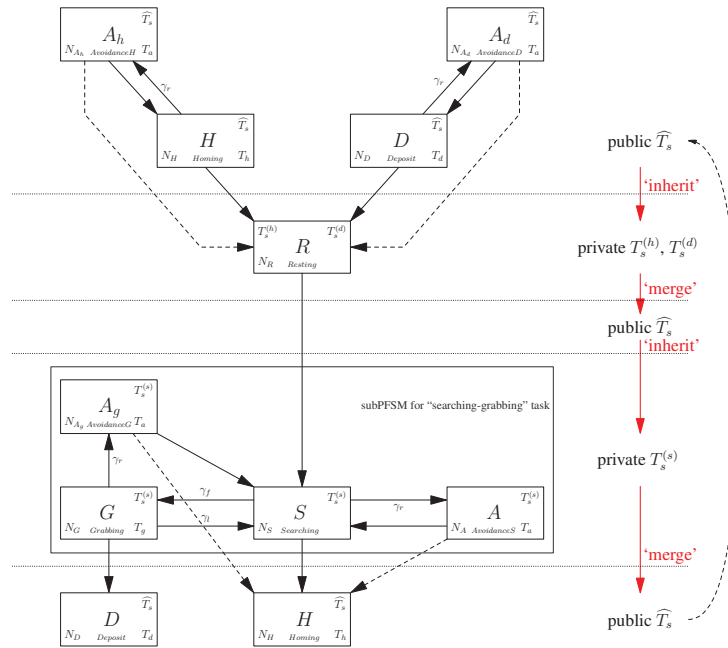


Figure 5: Relationship between private and public searching time threshold. The influence domain of each time threshold is separated with dotted lines. Private $T_s^{(h)}$ and $T_s^{(d)}$ coexist with private $T_r^{(h)}$ and $T_r^{(d)}$ (see Figure 4).

3.4.1 Social Cues

Again, let $T_s^{(h)}(k; i)$ and $T_s^{(d)}(k; i)$ be the private searching time threshold for the robots moving from state *Homing* and *Deposit* respectively, where i denotes the DOB (i.e. the time step that the robots move into state *Resting*) and k is the current time step, clearly $k \geq i$, then the mathematical description for $T_s^{(h)}$ and $T_s^{(d)}$ can be obtained using the same approach as presented in previous sections.

Initialisation

Initialisation is performed when the robots move into state *Resting*, i.e. ($k = i$). An ‘inherit’ operation is executed to make a copy of the current public searching time threshold, and then, based on the adaption rules, the private searching time threshold will be updated according to the pheromones left by the previous returning (to nest)

robots. Thus

$$T_s^{(h)}(k+1; k+1) = \widehat{T}_s(k) + \beta_1 SP_s(k) - \gamma_1 SP_f(k) \quad (25)$$

$$T_s^{(d)}(k+1; k+1) = \widehat{T}_s(k) + \beta_1 SP_s(k) - \gamma_1 SP_f(k) \quad (26)$$

Adaptation

After the robots transfer to state *Resting*, the social cues continue to adjust the private searching time thresholds, until the robots move to state *Searching*. The private searching time threshold $T_s^{(h)}$ and $T_s^{(d)}$ will be updated using the following rules

$$T_s^{(h)}(k+1; i) = T_s^{(h)}(k; i) + \beta_1 * \Delta_D(k - T_d) - \gamma_1 * \Delta_H(k - T_h) \quad (27)$$

$$T_s^{(d)}(k+1; i) = T_s^{(d)}(k; i) + \beta_1 * \Delta_D(k - T_d) - \gamma_1 * \Delta_H(k - T_h) \quad (28)$$

Although Equation (25) - Equation (28) show that the update rules for $T_s^{(h)}$ and $T_s^{(d)}$ are exactly the same, the life cycles for these two private searching time thresholds are different, as decided by the private resting time parameters $T_r^{(h)}$ and $T_r^{(d)}$.

Merging

The merging operation occurs when the resting robots run out of their resting time, decided by the private resting time thresholds $T_r^{(h)}$ and $T_r^{(d)}$. Let $\Delta_{T_s^{(h)}}(k)$ and $\Delta_{T_s^{(d)}}(k)$ represent the contribution provided by the robots transferred from state *Homing* and *Deposit* respectively, then

$$\Delta_{T_s^{(h)}}(k) = \sum_{i \in \mathbb{R}_H(k)} \Delta_H(i - T_h) \times (T_s^{(h)}(k; i) - \widehat{T}_s(i - 1)) \quad (29)$$

$$\Delta_{T_s^{(d)}}(k) = \sum_{i \in \mathbb{R}_D(k)} \Delta_D(i - T_d) \times (T_s^{(d)}(k; i) - \widehat{T}_s(i - 1)) \quad (30)$$

where $\mathbb{R}_H(k)$ and $\mathbb{R}_D(k)$ are collections of DOBs for the private resting time threshold which come to the end of their lifecycles, which are defined in Equation (15) and Equation (16). $\Delta_H(i - T_h)$ and $\Delta_D(i - T_d)$ depict the number of resting robots which are ready to transfer to state *Searching* at time step k .

Finally, the contribution of social cues to the public searching time threshold \widehat{T}_s can be expressed as

$$\widehat{T}_s(k+1) = \widehat{T}_s(k) + \frac{\Delta_{T_s^{(h)}}(k) + \Delta_{T_s^{(d)}}(k)}{N_0} \quad (31)$$

3.4.2 Environmental Cues

Once the robots move to state *Searching*, they are subject to the constraint of searching time threshold unless the robots grab food-items successfully. The environmental cues affect the searching time threshold when the robots are actively engaged in the searching task. To represent the unique and variable searching time threshold, a new private searching time threshold is introduced for the sub-PFSM engaged in ‘‘searching-grabbing’’ task, denoted $T_s^{(s)}(k; i)$, where i corresponds to the DOB of the private searching time threshold (and the sub-PFSM), and k is the current time step for the sub-PFSM. Similarly, we can define the ‘initialisation’, ‘adaptation’ and ‘merging’ operations for $T_s^{(s)}$.

Initialisation

Generally, when $T_s^{(s)}$ is formed, it should ‘inherit’ the up-to-date public searching time threshold. However, as shown in Figure 5, $T_s^{(h)}$ and $T_s^{(d)}$ are ‘merged’ to \widehat{T}_s at the same time. Thus the initialising of $T_s^{(s)}$ is the combination of both ‘merge’ and ‘inherit’ operations, i.e.

$$T_s^{(s)}(k+1; k+1) = \widehat{T}_s(k) + \frac{\Delta_{T_s^{(h)}}(k) + \Delta_{T_s^{(d)}}(k)}{N_0} \quad (32)$$

Clearly, the size of sub-swarm in the sub-PFSM equals the number of robots moving from state *Resting* currently, which can be expressed as

$$N_S'(k+1; k+1) = \Delta_{S \leftarrow R^h}(k) + \Delta_{S \leftarrow R^d}(k) \quad (33)$$

where $\Delta_{S \leftarrow R^h}(k)$ and $\Delta_{S \leftarrow R^d}(k)$ are defined in Equation (17) and Equation (18).

Adaptation

When $i < k < i + T_s^{(s)}(k; i)$, the change of $T_s^{(s)}$ can be described as follows

$$T_s^{(s)}(k+1; i) = T_s^{(s)}(k; i) + \frac{\alpha_1 * \Delta'_A(k+1)}{N'_S(i; i)} \quad (34)$$

where $\Delta'_A(k+1)$ depicts the number of robots transferring to state *avoidance* from state *Searching* in the sub-PFSM, $N'_S(i; i)$ is the initial number of robots in the sub-swarm, as defined in Equation (33).

Merging

Similarly, in order to know the contribution from the environmental cues during the lifecycles of the sub-PFSM, let $\mathbb{S}(k)$ denote the collection of all the DOBs for the sub-PFSMs which come to the end of their life cycles at time step k , then $\mathbb{S}(k)$ can be expressed as

$$\mathbb{S}(k) = \{i | k - i = \lfloor T_s^{(s)}(k; i) \rfloor\} \quad (35)$$

Whenever the robots in the sub-PFSM run out of their searching time, their private searching time threshold $T_s^{(s)}$ will be 'merged' to the public searching time threshold in this way

$$\widehat{T}_s(k+1) = \widehat{T}_s(k) + \frac{\sum_{i \in \mathbb{S}(k)} [T_s^{(s)}(k; i) - T_s^{(s)}(i; i)] * N'_S(i; i)}{N_0} \quad (36)$$

3.4.3 Combining all the Cues

As the social and environmental cues may occur simultaneously in the swarm, we need to merge Equation (31) and (36) to model the effect of the social cues and environmental cues on the public searching time threshold. Thus we have

$$\widehat{T}_s(k+1) = \widehat{T}_s(k) + \frac{\sum_{i \in \mathbb{S}(k)} [T_s^{(s)}(k; i) - T_s^{(s)}(i; i)] * N'_S(i; i) + \Delta_{T_s^{(h)}}(k) + \Delta_{T_s^{(d)}}(k)}{N_0} \quad (37)$$

3.5 Integration with Previously Developed Model

To obtain a complete model of adaptive collective foraging, we need to integrate this work into our previously developed model in [2]. This can be done by replacing some equations with the new working as follows:

$$N_S(k+1) = N_S(k) + \Delta_S(k+1) + \gamma(k)N_G(k) + [\Delta_A(k - T_a) - \Omega_A(k - T_a)] + [\Delta_{A_g}(k - T_a) - \Omega_{A_g}(k - T_a)] - [\gamma_r(k) + \gamma_f M(k)]N_S(k) - \Gamma_S(k+1) \quad (38)$$

$$N_R(k+1) = N_R(k) + \Delta_D(k - T_d) + \Delta_H(k - T_h) - \Delta_S(k+1) \quad (39)$$

$$\Gamma_S(k) = \sum_{i \in \mathbb{S}(k)} N'_S(k; i) \quad \Gamma_G(k) = \sum_{i \in \mathbb{S}(k)} N'_G(k; i) \quad (40)$$

$$\Gamma_A(k) = \sum_{i \in \mathbb{S}(k)} N'_A(k; i) \quad \Gamma_{A_g}(k) = \sum_{i \in \mathbb{S}(k)} N'_{A_g}(k; i) \quad (41)$$

$$\Omega_G(k - T_g) = \sum_{k'=k-T_g}^k \sum_{i \in \mathbb{S}(k')} \Delta'_G(k - T_g; i) \quad (42)$$

$$\Omega_A(k - T_a) = \sum_{k'=k-T_a}^k \sum_{i \in \mathbb{S}(k')} \Delta'_A(k - T_a; i) \quad (43)$$

$$\Omega_{A_g}(k - T_a) = \sum_{k'=k-T_a}^k \sum_{i \in \mathbb{S}(k')} \Delta'_{A_g}(k - T_a; i) \quad (44)$$

4 Results and Conclusion

The extended macroscopic model has been validated using the sensor-based simulation tools Player/Stage (a screen shot is shown in Figure 1). The basic parameters for the simulation environment, for instance the size of arena, the speed of the robots, etc, are exactly the same as were used in [2]. The behaviour sets of the robots in the simulation are also the same, with an exception that each robot is now endowed with the adaptation ability. Using the same set of adjustment factors presented in [1], we have also tested the model with different food growth rates (i.e. the probability that one food item *grows* in the arena, each second). Figure 6 illustrates the results from both the simulation and macroscopic probabilistic model for a swarm of 8 robots, where the growth rate varies from 0.03 to 0.05. The error bars represent the standard deviations of data recorded from 10 experimental runs. We see clearly that the data from simulation fits well to the curves obtained from the macroscopic model, though a relatively large gap develops when the growth rate is set to 0.03. Figure 7 then plots the instantaneous number of robots in selected states from the simulation under different environmental conditions. Not surprisingly, the predicted number of robots in each state from the macroscopic model reflects the corresponding average number of robots from the

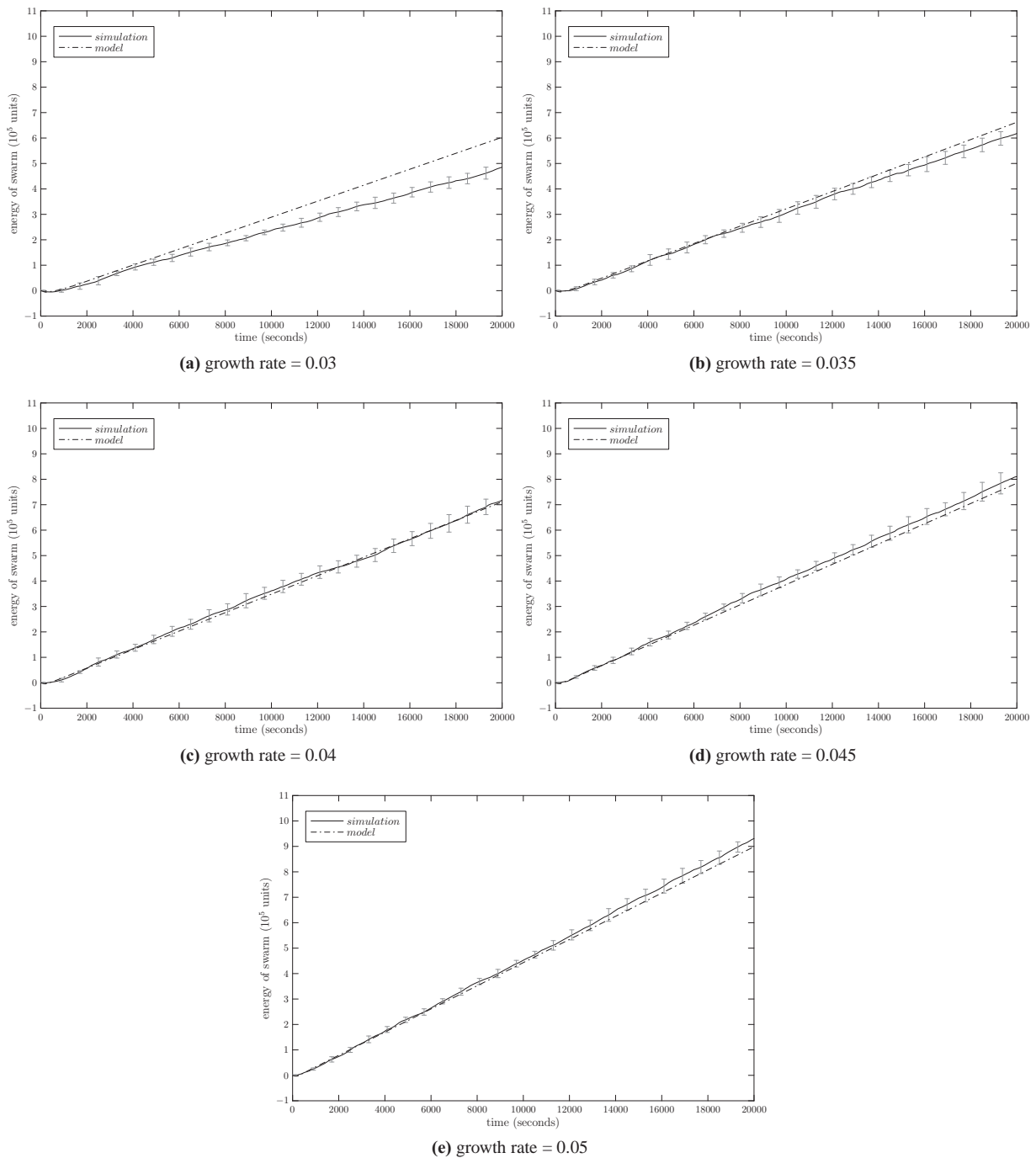


Figure 6: Comparison of the instantaneous energy of the swarm with adaptation mechanism between the extended macroscopic probabilistic model and the simulation.

simulation. As we already discussed in [1] and [2], for a fixed population swarm under different environmental conditions – i.e. food growth rate, there is a different optimal swarm density (ratio of foragers to search area) which results in the optimal energy efficiency for the system. Although the swarm population and the adjustment factors remain the same for the above experiments, the average number of robots in state *Resting* (referred to as resters) and non-*Resting* (as foragers) vary with the food growth rate changing. This indicates that the swarm with adaption is able to adjust the ratio of foragers and resters autonomously in response to environmental changes. In other words, the swarm density (in foragers) in the arena varies with food growth rate varying.

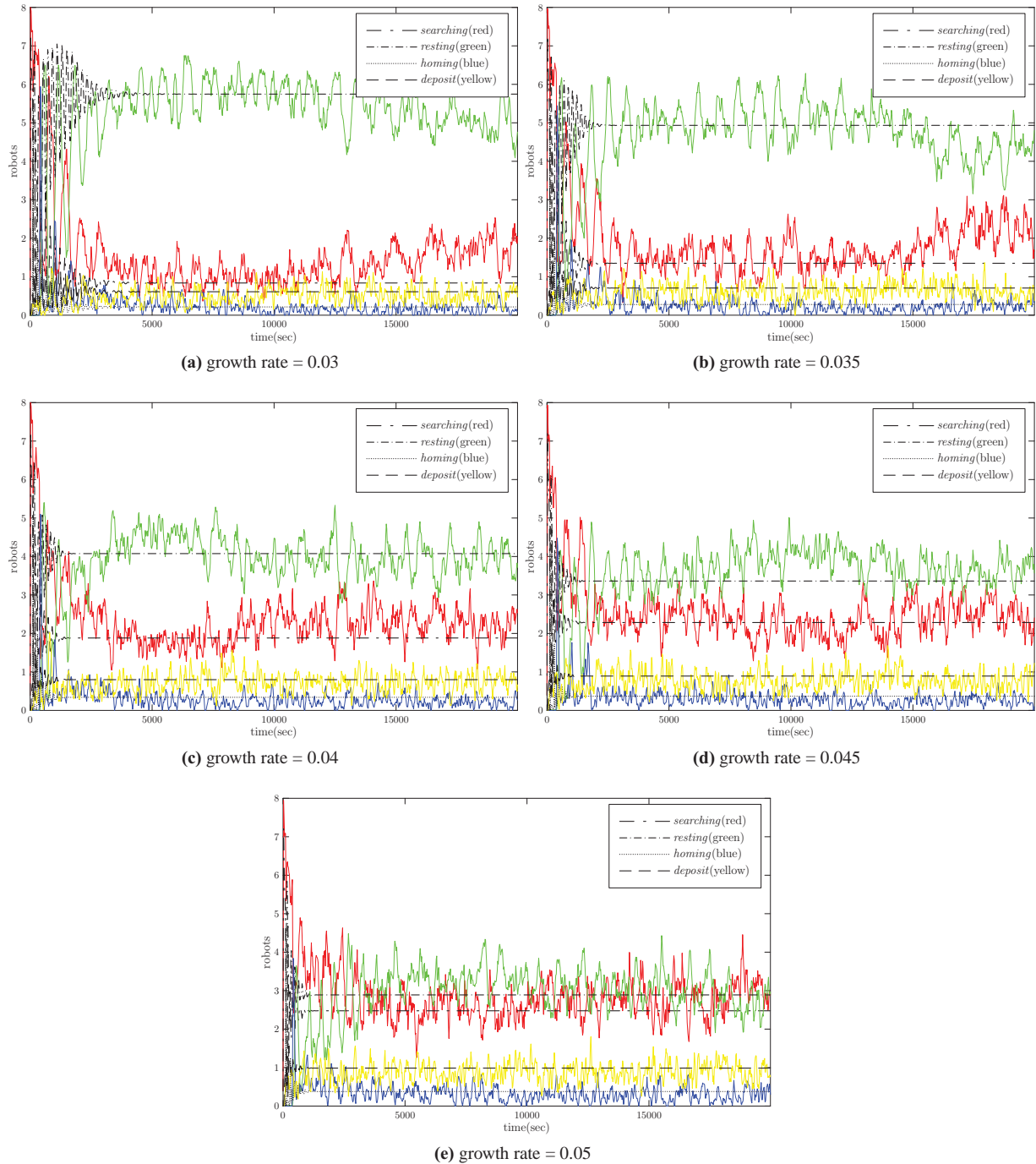


Figure 7: Comparison of number of robots in selected states for the swarm with adaptation mechanism between the extended macroscopic probabilistic model and the simulation, where the horizontal dashed lines are from the model while the coloured curves from simulation.

Clearly, by setting all the adjustment factors to zero, the extended model should reduce to be the simplified case we developed in [2]. To examine this, let us consider the situation where the growth rate is 0.045 for a swarm of homogeneous robots (without adaptation), we can derive the relationship between the resting time threshold and the final net energy of the swarm by varying the resting time threshold T_r from 0 to 1000 time steps (corresponding to 0 to 200 seconds at 5 Hz updating rate). Figure 8 compares the results obtained from both the simulation and the

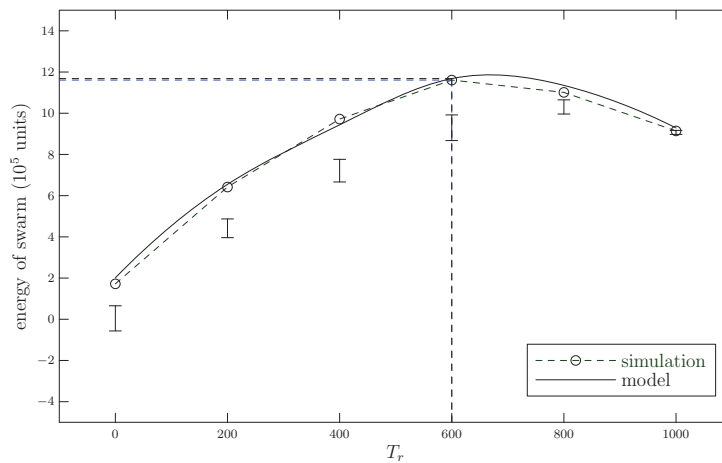


Figure 8: The total net energy of the swarm after 20000 seconds for a swarm of 8 robots with different resting time threshold parameter.

macroscopic model. Again, the final net energies of the swarm predicted by the model match with those measured from simulation very well. We also see that a critical value of T_r exists, corresponding to the maximum net energy of the swarm. As for a fixed population of robot swarm, it is understood that the bigger the resting time threshold T_r , the larger the proportion of robots stay in state *Resting* on average, as a consequence the swarm density (in foragers) in the arena is smaller.

Due to the huge solution space of the adjustment factors, it is not possible to test the model by varying each parameter individually and repeating the experiment again and again. Although no further comparisons are made at this stage, as the adjustment factors are chosen intuitively, we have good reason to believe that the model developed in this paper truly captures the dynamics of the swarm with adaptation. Clearly the swarm with intuitively selected adjustment factors does not reach its optimal performance if we compare the results presented in Figure 6 and Figure 8. However, in conjunction with an appropriate searching technique, the macroscopic model can be used to find an optimal set of adjustment factors for the adaptation algorithm and hence help the swarm achieve the best performance; we have implemented this approach using a genetic algorithm in [26]. To the best of our knowledge, at the time of writing, there are very few macroscopic models in the field of swarm robotics that can describe the collective behaviour of a group of heterogeneous robots. Although the model presented in this paper is specific to the adaptive foraging task, we believe the methodology can be extended to other systems in which the heterogeneity of the system is coupled with its time parameters.

5 References

- [1] Wenguo Liu, Alan F. T. Winfield, Jin Sa, Jie Chen, and Lihua Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305, 2007.
- [2] Wenguo Liu, Alan F. T. Winfield, and Jin Sa. Modelling swarm robotic system: A case study in collective foraging. In Myra S. Wilson, Frédéric Labrosse, Ulrich Nehmzow, Chris Melhuish, and Mark Witkowski, editors, *Proceeding of Towards Autonomous Robotic Systems*, pages 25 – 32, Aberystwyth, UK, September 2007.
- [3] Alcherio Martinoli, Auke Jan Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.
- [4] Alcherio Martinoli, Auke Jan Ijspeert, and Luca Maria Gambardella. A probabilistic model for understanding and comparing collective aggregation mechanisms. In *ECAL '99: Proceedings of the 5th European Conference on Advances in Artificial Life*, pages 575–584, London, UK, 1999. Springer-Verlag.
- [5] D. J. T. Sumpter and S. C. Pratt. A modelling framework for understanding social insect foraging. *Behavioral Ecology and Sociobiology*, 53:131–144, 2003.
- [6] Ken Sugawara and Masaki Sano. Cooperative acceleration of task performance: foraging behavior of interacting multi-robots system. *Phys. D*, 100(3-4):343–354, 1997.
- [7] K. Sugawara, M. Sano, I. Yoshihara, K. Abe, and T. Watanabe. Foraging behaviour of multi-robot system and emergence of swarm intelligence. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 3, pages 257–262 vol.3, 1999.
- [8] K. Sugawara and T. Watanabe. A study on foraging behavior of simple multi-robot system. In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, volume 4, pages 3085–3090 vol.4, 2002.
- [9] Kristina Lerman and Aram Galstyan. A general methodology for mathematical analysis of multi-agent

- systems. Technical report, 2001.
- [10] Kristina Lerman and Aram Galstyan. Two paradigms for the design of artificial collectives. In *workshop on Collectives and Design of Complex Systems, NASA/Ames*. Springer-Verlag, 2002.
 - [11] Kristina Lerman. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.
 - [12] Kristina Lerman, Aram Galstyan, and Tad Hogg. Mathematical analysis of multi-agent systems. 2003.
 - [13] W. Agassounon and A. Martinoli. A Macroscopic Model of an Aggregation Experiment using Embodied Agents in Groups of Time-Varying Sizes. In *Proc. of the IEEE Conf. on System, Man and Cybernetics SMC-02*, 2002.
 - [14] Alcherio Martinoli. Optimization of swarm robotic systems via macroscopic models. In *Proc. of the Second Int. Workshop on Multi-Robots Systems*, pages 181–192, 2003.
 - [15] Alcherio Martinoli and Kjerstin Easton. Modeling swarm robotic systems. *Springer Tracts in Advanced Robotics*, 5:297–306, 2003.
 - [16] Alcherio Martinoli, Kjerstin Easton, and William Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *International Journal of Robotics Research*, 23(4):415–436, 2004.
 - [17] Nikolaus Correll and Alcherio Martinoli. Modeling and Analysis of Beaconless and Beacon-Based Policies for a Swarm-Intelligent Inspection System. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2488–2493, 2005.
 - [18] N. Correll and A. Martinoli. Modeling and analysis of beaconless and beacon-based policies for a swarm-intelligent inspection system. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2477–2482, 18-22 April 2005.
 - [19] Nikolaus Correll and Alcherio Martinoli. Towards Optimal Control of Self-Organized Robotic Inspection Systems. In *8th International IFAC Symposium on Robot Control (SYROCO)*, 2006.
 - [20] Nikolaus Correll and Alcherio Martinoli. Modeling and Optimization of a Swarm-Intelligent Inspection System. In *Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, Distributed Autonomous Robotic Systems, Springer Verlag, pages 369–378, 2006.
 - [21] Nikolaus Correll and Alcherio Martinoli. Modeling Self-Organized Aggregation in a Swarm of Miniature Robots. In *IEEE 2007 International Conference on Robotics and Automation Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*, 2007.
 - [22] Kristina Lerman and Aram Galstyan. Macroscopic analysis of adaptive task allocation in robots. *Proceeding of International Conference on Intelligent Robots and Systems (IROS-03)*, 2003.
 - [23] Aram Galstyan and Kristina Lerman. Analysis of a stochastic model of adaptive task allocation in robots. In *Agent Modeling Workshop, AAMAS-04*, 2004.
 - [24] Kristina Lerman, Chris Jones, Aram Galstyan, and Maja J Matarić. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 25(3):225–242, 3 2006.
 - [25] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, 2003.
 - [26] Wenguo Liu. *Design and Modelling of Adaptive Foraging in Swarm Robotic Systems*. PhD thesis, University of the West of England, 2008.