Net Energy Income Optimization in a Homogeneous Swarm Robotics Foraging System

Nizar H. Abbas and Prof. Rameshwar Rao

Department of Electronics & Communication Engineering
Osmania University, Hyderabad, Andhra Pradesh, India
E-mail: rniy797506@yahoo.com

Abstract

This paper presents an optimization technique inspired from the collective foraging observed in natural insects, swarm robotics is a new approach to coordinate the behaviors of large number of relatively simple robots in decentralized manner. In such robotic systems, an individual robot have only local perception and very limited capabilities in terms of sensing, computation, and communication can adapt its own behavior so that a desired collective behavior emerges from the local interactions among robots and between robots and the environment. Swarm robotics has been the focus of increased attention recently because of the beneficial features demonstrated in such systems, such as higher swarm efficiency, robustness against the failures of individual robots, flexibility to adapt to changes in the environment, and scalability over a wide range of swarm sizes. In this paper we present an optimization technique to regulate the net energy income of an individual robot performing collective foraging tasks. Through the interactions between robots, a desired division of labour can be achieved at the swarm level. Robot swarm also demonstrates the ability to optimize energy efficiency and its potential robustness in different environments.

Keywords: Swarm robotics, collective foraging, net energy, energy optimization, homogeneous system.

Introduction

Inspired by the swarm intelligence observed in social insects, robotic swarm are fully distributed systems in which overall system tasks are typically achieved through self-organization or emergence rather than direct control [1]. In swarm robotics a number of relatively simple robots, each with limited sensing, actuation and cognition, work
together to collectively accomplish a task. Such tasks may be biologically plausible, such as cluster sorting [2] or cooperative stick-pulling [3]. Foraging, however, is a compelling example of a swarm behavior that can be transferred from natural to artificial systems because of the one-to-one correspondence between ant and robot and between food-item and energy units. Foraging can, in principle, be undertaken by a single robot given enough time, but a swarm of robots working together should be able to complete the task more quickly and effectively. Foraging is, therefore, an example of a swarm robotics behavior in which it is not the task itself, but the way the task is self-organized, that is the desired emergent property of the swarm.

In this paper, we investigate the problem of homogeneous swarm robots performing a collective foraging task. Collective foraging is a research problem which has often been used in multi-robotics system design. In a collective foraging scenario, a swarm of robots has to search for objects called “food” that are randomly scattered in a restricted place called the “foraging area.” Once it finds food, the robot will take the food back to a certain designated place called “home.” Collective foraging is often used as a model for a wide range of real-life applications, such as toxic-waste cleanup, search and rescue, and collection of terrain samples in unknown environments [4].

It is a characteristic of foraging that the foraging robots for food will acquire energy from the food they retrieve but will also expend energy on motion during the foraging process. Net energy is the total energy acquired, less the energy cost by the swarm. The main concern in this study is to determine whether robots are able to cooperate in order to optimize the net energy in a timely manner and also adapt to unknown changes in the environment. Each robot used in this study has only limited capabilities in terms of sensing, computational power, and communication. Due to these limits, a single robot is not capable of knowing the global state of the environment or the overall task progress.

There are several considerations that could be taken into account in order to improve the net energy of the swarm. The first consideration is the number of foraging robots. If this number is too high, since the robots are foraging in a bounded foraging area, interference among the robots not only costs more energy but also decreases the probability of a robot finding food. In addition, more robots actively foraging costs more energy. On the other hand, if the number of actively foraging robots is low, the swarm might not retrieve enough food energy from the environment. Therefore, there is an optimal value for the number of active foraging robots in a given environment that should optimize the net swarm energy. In order for a robot swarm to be robust and flexible, this optimal value should be able to adapt to a new value in the changing foraging environment.

The early research in collective foraging field [5, 6] focused on the use of communication design to assess spatial characteristics of the foraging environment in order to coordinate the robots to fulfill the task. Beacon-following methodology characterized the earliest efforts at collective foraging. Researchers have also understood the effectiveness of trail-laying and following in the foraging strategies of social insects, such as ants [7], which provide inspiration for swarm robotic foraging.

Nowadays, there has been increased work in investigating the division of labour technique in collective foraging. Division of labour here means the division of the
tasks of actively foraging and resting at “home” among robots in the swarm so that
the net energy income of the swarm can be optimized. Our motivation for using a
division of labour comes directly from the behavior of social insects that we observed
in nature. Krieger and Billeter [8] implemented a swarm of up to twelve real robots to
demonstrate the efficiency of self-organized task allocation in the performance of a
collective foraging task. Labella et al. [9, 10] introduced a simple adaptive mechanism
to change the ratio of foragers to resters in order to improve the swarm foraging
labour between collections of two different objects. Guerrero and Oliver [12] present
an auction-like task allocation model, trying to specify the optimum number of
foragers.

Model of a Swarm Robotic Foraging System
The main contribution of this paper is identifying the individual robot behavior that
could lead to efficient swarm robotic foraging behavior. We assume this is
homogeneous system; all robots would follow the same rule. The control algorithm
gets inspiration from the mechanism of labour division in social insects. It enables the
robotic system to achieve a desired division of labour over a swarm of robots in order
to achieve higher swarm energy efficiency. This division of labour also needs to be
dynamically adjusted in response to the changes in environment stimulus or
individual foraging performance.

The finite state machine (FSM) in Fig.1 illustrates the different states of collective
foraging behavior of individual robot in our study. Transitions between states occur
on the base of events that are either external (e.g. success retrieval or fail) or internal
of the robot (e.g. deposit). The states for robot foraging behavior are as follows:

a) Homing: In this state the robot must move, with its collected object, to a home
or nest region. Homing clearly requires a number of stages, firstly,
determination of the position of the home region relative to where the robot is
now, secondly, orientation toward that position and, thirdly, navigation to the
home region. Again there are a number of strategies for homing: one would be
to re-trace the robot’s path back to the home region using, for instance,
odometry or by following a marker trail; another would be to home in on a
beacon with a long range beacon sensor. When the robot has successfully
reached the home region it will change state to depositing.

b) Searching: In this state the robot is physically moving through the search
space using its sensors to locate and recognize the target items. At this level of
abstraction we do not need to state how the robot searches: it could, for
instance, wander at random, or it could employ a systematic strategy such as
moving alternately left and right in a search pattern. The fact that the robot has
to search at all follows from the pragmatic real-world assumptions that either
the robot’s sensors are of short range and/or the items are hidden (behind
occluding obstacles for instance); in either event we must assume that the
robot cannot find items simply by staying in one place and scanning the whole
environment with its sensors. Object identification or recognition could
require one of a wide range of sensors and techniques. When the robot finds an item it changes state from searching to grabbing. If the robot fails to find the target item then it remains in the searching state forever; searching is therefore the ‘default’ state.

c) Depositing: In this state the robot deposits or delivers the item in the home region, and then immediately changes state to searching and hence resumes its search.

**Figure 1:** FSM of Collective Foraging Behaviour

The transitions between states occur on a basis of events that are either external (e.g., food located or time out) or internal to the robot (e.g., deposit). The transitions between the above states are explained as follows:

a) Start Foraging: Robot leaves home, starts foraging.

b) Success Retrieval: If a robot finds object, it grasps the object.

c) Time Out: If the energy of a robot is used up while the robot still searching, the robot gives up foraging and goes home (failed retrieval).

d) No Object: Robot has not found objects and keeps searching and interacting.

In order to regulate robot behavior so that a beneficial division of labour can be achieved, we introduce the variable foraging probability $P$ for each robot. For example, $P(k)$ is the foraging probability of robot $k$ for which $k$ is the robot ID. Only when $P(k)$ is higher than the threshold value $P_0$, will robot $k$ start foraging; otherwise, robot $k$ will rest at home. Two variables were used to calculate $P(k)$: foraging threshold and foraging stimulus $F_s$. $F_s(k)$ relates to the foraging performance of robot $k$ and $F_s$ represents the foraging task stimulus for the swarm. The mathematical model we use here to calculate $P(k)$, as shown in Eq.(1), can be considered an instance of a response threshold model as presented in Bonabeau et al., [13] Thiraulaz et al., [7] and Labella, [10] In order to explain the division of labour in social insects, Bonabeau has developed a model that relies on response thresholds for
each individual. In his model, every individual has a fixed response threshold for every task. Individuals engage in task performance only when the level of task-associated stimulus exceeds their threshold. When individuals with a lower response threshold for performing a given task are withdrawn from the swarm, less task-related work will be done and the intensity of the stimulus is increased. The stimulus eventually will reach the high response thresholds of the remaining individuals. Theraulaz [14] has extended the fixed threshold model by allowing a threshold to vary in time, following a simple reinforcement process: a threshold decreases when the corresponding task is performed and increased when the corresponding task is not performed. The more an individual performs a task, the lower the response threshold, and vice versa.

\[ P(k) = \frac{F_s^2}{F_s^2 + F_T^2(k)} \]  

(1)

Fig. 2 shows how two variables, \( F_T(k) \) and \( F_s \), relate to the \( P(k) \) from the Eq.(1). The plot is a series of probability curves according to the equation. Each curve has a fixed value of \( F_T(k) \) along changing \( F_s \). The graph shows that, with a fixed value of \( F_T(k) \), one will have a higher foraging probability \( P(k) \) when stimulus \( F_s \) increases. Under the same value of \( F_s \) on different curves, a robot with a lower threshold has a higher foraging probability.

**Figure 2** : Probability of Foraging Response with respect to Stimulus with Different Foraging Threshold

The division of labour mechanism in this work is inspired by the above response threshold model. It considers both the foraging threshold of an individual robot and environment-related stimulus intensity. We proposed two algorithms to adjust \( F_T(k) \) and \( F_s \) one at searching arena and the other algorithm at home so that the
number of actively foraging robots can be adjusted accordingly.

The two algorithms are explained as follows:

a. Algorithm No.1 in Searching Arena: when foraging robot \( k \) encounters another foraging robot \( l \), it exchanges foraging state information with robot \( l \) and records the foraging state of robot \( l \) in a task counter of foraging robot \( k \). The algorithm are:

\[
\text{Algorithm 1: Adjusting } F_T(k) \text{ and } F_S \text{ in Searching Arena}
\]

\[
\begin{align*}
\text{Initialization:} \\
& \text{foraging robot ID} = R_{id} \\
& \text{if } R_i \text{ has already found object, then} \\
& \quad \text{Taskounter}(k) \rightarrow \text{TaskCounter}(k)+1 \\
& \text{else if } R_i \text{ is in searching state, then} \\
& \quad \text{Taskounter}(k) \rightarrow \text{TaskCounter}(k)-1 \\
& \text{else if } R_i \text{ is in fail state, then} \\
& \quad \text{Taskounter}(k) \rightarrow \text{TaskCounter}(k)-2 \\
& \text{end if}
\end{align*}
\]

As long as robot \( k \) is moving in the foraging field, it keeps interacting and collecting information from other robots. TaskCounter(\( k \)) accumulates the information it collects.

b. Algorithm No.2 at Home: Once robot \( k \) reaches home, it calculates the net energy from the aging trip to see if it is positive or negative. Positive net energy means successful foraging it will decrease its own foraging threshold \( F_T(k) \) by factor of \( \nabla \) and increasing the global foraging task stimulus \( F_S \) by factor of \( \varsigma \) (i.e., if I am successful, I will increase the probability of foraging again by lowering the threshold and also if I am successful and most other robots that I encountered are successful, there must be a lot of object out there, which increase the stimulus of foraging). Negative net energy means failed foraging and that will increase its own foraging threshold \( F_T(k) \) by factor of \( \nabla \) and decreasing the global foraging task stimulus \( F_S \) by factor of \( \varsigma \) (i.e., if I fail, I will decrease the probability of foraging again by increasing the threshold and also if I failed and most other robots that I encountered are still searching or failed, it seems there is not much object left, so I will decrease the stimulus of foraging).
Algorithm 2: Adjusting $F_T(k)$ and $F_S$ at Home

Initialization:
foraging robot ID = $R_{ID}$
if $R_i$ success and TaskCounter$(k) > 0$, then
  $F_T(k) \rightarrow F_T(k) - \nabla_1$
  $F_S \rightarrow F_S + \varsigma_1$
else $R_i$ failed and TaskCounter$(k) < 0$, then
  $F_T(k) \rightarrow F_T(k) + \nabla_2$
  $F_S \rightarrow F_S - \varsigma_2$
end if

With these algorithms described above, each robot in the swarm will adjust its own foraging probability $P(k)$ according to updated $F_T(k)$ and $F_S$. At the swarm level, the number of active foragers can be automatically adjusted so that swarm energy efficiency can be optimized.

The computer simulator which we have used is the Webots, a 3D sensor-based, kinematics to validate the algorithms presented in the previous section. The Webots simulator simulates a population of twelve mobile robots foraging for food in a two-dimensional foraging arena.

**Experimental setup**

**Simulation environment**
To validate the algorithms presented in the previous section, we tested our swarm foraging using the sensor-based simulation Webots. Twelve robots work in an $8m \times 8m$ octagonally shaped arena. An advantage of this approach is that the Webots control program can be transferred directly from simulation to real robots.

A set of experiments was designed which varied the size of the robot swarm and food density in the foraging area. Two criteria are designed for each type of experiment. In criterion $C_1$, in which no algorithms are used, the system randomly chooses another robot to forage when one robot returns home. The number of active foragers remains at the same value as the swarm size during the simulation. This provides us a benchmark for comparison. Criterion $C_2$ uses the proposed two algorithms to update $F_T(k)$ and $F_S$. The number of active foragers will change over time accordingly.

**Selection of Parameters**

Before starting the experiments, we first need to choose values for the parameters and initial settings. The factors $\nabla_1, \nabla_2, \varsigma_1, \varsigma_2$ are applied to update $F_T(k)$ and $F_S$. The
selection of values of these parameters is based on trial and error. In order to calculate net energy income, we assume one food-unit can bring 4000 units of energy back to the swarm and the robot would expend sixteen units of energy per second while moving and another fifteen energy units every time it encounters and avoids another robot. In order to return a positive net energy value back to the swarm, we set a searching energy limit for each robot of 3800 units Table 1 summarizes all of these parameters we have chosen for the experiments.

Table 1: Experiments Parameters

<table>
<thead>
<tr>
<th>( \nabla_1 )</th>
<th>( \nabla_2 )</th>
<th>( \varsigma_1 )</th>
<th>( \varsigma_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
</tr>
</tbody>
</table>

The equation we use to measure the net energy efficiency of the swarm robotic foraging system, called “swarm energy efficiency,” is given below. In this definition, food energy available from the environment is food energy put in the arena by the system over the simulation time.

\[
\text{Swarm Energy Efficiency} = \frac{\text{Net Energy Income to Swarm}}{\text{Food Density Available in Environment}}
\] (2)

Experimental Result and Analysis

Variable Swarm Size, Fixed Food Density

To explore the optimal number of foragers, we ran experiments for the swarm sizes of 2, 4, 6, 8, 10, and 12 robots, respectively. The food density \( F_d \) in the experiments is fixed to 2/min, which means the system will place two food-units in the arena every minute. For each swarm, we apply two criteria and each simulation lasts 200 minutes, which means there are a total of 400 food-units in the arena. We record the number of food-units collected, the number of active foragers, the values of the stimuli and the net swarm energy.

We compare the food collect rate of the swarm using different criteria. Our data show that most swarm can reach a high collect rate with the exception of the swarm with 2 or 4 robots. For the swarm with 6, 8, 10, and 12 robots, more than 95% of the food is collected no matter which criterion is used. However, the swarm with 2 and 4 robots collects less than 90% of the food in most cases. Therefore, in order to collect the most food in a given environment, more than 4 robots are needed in active foraging.

Checking the average number of active foragers using criterion \( C_1 \) for these experiments, we find the averages are all close to 6. All robots are foraging most of the time in the swarm with 2 and 4 robots since there is enough food available for retrieval and for the swarm of 12 robots with criterion \( C_2 \), the average number of foragers is 6.2. In other words, more than 4 robots need to be engaged in foraging in order to collect all the food; meanwhile, the more robots resting at home, the more
energy the swarm can save. Therefore in the given foraging environment, in order to maximize the net energy income for the swarm, the optimal average number of active foragers is close to 6. Here, foraging probability in $C_3$ helps a robot to switch tasks between foraging and resting more effectively, allowing the number of active foragers to reach the optimal value. Thus, the overall division of labour in a swarm emerges from the low-level interactions between robots and the environment.

We compare the energy efficiencies of the swarm using different criteria. The efficiency levels are nearly the same in the swarm with 4 robots or fewer, since all robots are engaged in foraging. However, for the swarm with more than 4 robots, the swarm with criterion $C_2$ can always obtain a higher energy efficiency. Fig. 3 plots the instantaneous net swarm energy along with time. The net energy gap between criterion $C_1$ and $C_2$ increases as the swarm size grows. Thus we can conclude that, for a large swarm population, the proposed algorithms will not only help the swarm achieve a division of labour among robots but also will guide the swarm toward net energy income optimization in a given environment.

![Simulation Result](image)

**Figure 3:** Simulation Result for Different Swarm Sizes and Different Criteria with $F_d=2/min$
Fixed Swarm Size, Variable Food Density

We designed a second set of simulations to investigate how the proposed algorithms can help swarm robotic foraging under different environmental conditions; here we fix the size of the swarm to 12 robots but run the simulations with three different food source densities, from poor ($F_d = 1$/min) to relatively rich ($F_d = 4$/min). Two criterion $C_1$ and $C_2$ are used for the same swarm. Data from the simulations are recorded and plotted in Fig. 4.

\[ \text{Swarm Robotic Foraging Efficiency} \]

\[ \text{Food Density Rate} \]

**Figure 4 : Simulation Result for Same Swarm Size and Different Environment and Criteria**

The food collecting rate is >95% no matter what the environment might be, since there are enough robots for foraging. Fig. 5 plots swarm energy efficiency changes in different foraging environments. The swarm robotic foraging with criterion $C_2$ always has higher net energy efficiency, while the group using criterion $C_1$ is less efficient in all experiments. The gap between the two criteria becomes smaller in an environment with a higher food density. Despite the food source difference, the levels of energy efficiency for the groups with criterion $C_2$ are quite stable over different food sources, when compared criterion $C_1$, which implies that the swarm with the two algorithms is quite robust to environmental changes.

Since we used the same swarm in different foraging environments, stimulus ($F_S$) value indicates stimulus intensity from the environments. Fig. 4 shows that, in a richer environment ($F_d = 4$/min), the swarm has the highest $F_S$, and in a poor environment ($F_d = 1$/min) swarm has the lowest $F_S$. This means that through interactions among robots, a swarm can collectively perceive information about food sources in a foraging environment.
The swarm exhibits the capacity to perceive the environment collectively if we take into account the average number of active foragers over time. That is, more active foraging robots indicate a richer food environment and more inactive robots indicate a poor food environment. The average number of active foragers under different environments is plotted in Fig. 6. The average number of active foragers in the group using criterion $C_2$ is smaller when the food source becomes poorer and bigger when environment become richer. Individual robots cannot know global information about food sources in the environment; this correlation can only be observed at the overall swarm level and cannot be deduced from individual robots.
Conclusions

In this paper, we have proposed a simple two algorithm for a swarm robotic foraging which able dynamically change the number of foragers and thus make the swarm more net energy income optimized. Robot in the swarm modifies its foraging probability based on foraging performance (successful or failed food retrieval) and environment related stimulus through locally perceived information (collision with other robots while searching). The division of labour technique has been achieved in the swarm. Some robots will rest in home for longer to either save energy or minimize interference, or be actively engaged in foraging (which costs more energy for the individual but potentially gains more energy for the swarm).

With the purposed two algorithms, the swarm robotic foraging demonstrates:
   a. Significantly Improved swarm net energy income
   b. Division of labour (dynamic task allocation) between foraging and resting
   c. Robustness to environmental change (in food density)

Furthermore, the swarm with the two proposed algorithms seems to be able to guide the system towards energy optimization despite the limited sensing abilities of the individual robots and the simple social interaction rules. The swarm also exhibits the capacity to perceive the environment collectively if we take into account the average number of active foragers in the swarm over time. That is, more active robots indicate a richer food environment and more inactive robots indicate a poor food environment. This correlation can only be observed at the overall swarm level and cannot be deduced from individual robots.

This study is ongoing, and thus far we have tested our approach in swarms with 12 robots but we have not tested the scalability of the approach to swarms with hundreds or thousands of robots; however, given the minimal local communication between robots we have good reason to suppose the approach is scalable. We also have confidence that the approach will exhibit a high level of robustness to failure of individual robots, in keeping with the levels of robustness commonly seen in swarm robotic systems.

References


