Maximum Sustainable Yield Problem for Robot Foraging and Construction System

Ruohan Zhang and **Zhao Song**
Department of Computer Science, University of Texas at Austin, Austin, United States
{zharu,zhaos}@utexas.edu

Abstract

We introduce the Maximum Sustainable Yield problem for a multi-robot foraging and construction system, inspired by the relationship between the natural resource growth and harvesting behaviors in an ecosystem. The resources spawn according to the logistic model and are vulnerable to overharvesting. The robots must maintain sustainability while maximizing productivity. The foraging robots harvest different types of resources, which enable a construction robot to build new foraging robots. We design algorithms to perform robot construction, assignment, and scheduling. We propose an adaptive algorithm to overcome the problem that resource growth model is often unknown. We demonstrate that our algorithms are robust to noises in the actuation and the environment. The case where the observation noise could harm sustainability is discussed.

1 Introduction

In a multi-robot foraging problem, a team of robots harvest natural resources, such as fish, lumber, or minerals. This simple framework could extend to robot cleaning, search and rescue, landmine clearance, and planetary exploration [Winfield, 2009]. This paper addresses the environmental *sustainability* issue. The natural resource growth models vary from a simple constant growth model to the Malthusian Model [Malthus, 1872] or a more sophisticated logistic model [Verhulst, 1838]. A challenging goal is to maintain ecological balance of a vulnerable resource base while achieving productivity with a logistic growth model. This optimal harvest rate is known as the Maximum Sustainable Yield (MSY) [Hjort *et al.*, 1933]. In this paper, we design and implement control algorithms for a multi-robot foraging system to achieve MSY.

Several issues of [Song and Vaughan, 2013]'s work that introduced the multi-robot MSY foraging problem remain to be solved. First, it assumes that the logistic growth model parameters are known, so that optimal number of foraging robots can be calculated beforehand and the problem becomes task assignment. In reality, the parameters for a particular natural resource base is often unknown. Although the foraging industry has obtained sophisticated population growth models, the model parameters differ across natural resource bases so a single set of parameters will not generalize. Explicitly estimating these parameters is not trivial [Oliver, 1964] and could be impractical for three reasons. First, resource population is often at equilibrium without interference, which does not provide useful information for estimating the model. Second, some resources grow slowly hence data collection for estimating the model is slow. In the fishery and logging industry, it might take months or years for fish or wood to grow. Perhaps the most practical reason is simply the profit: the foraging companies often start harvesting immediately upon discovering a resource base. For these reasons, we analyze the mathematical properties of the logistic growth model, and develop an adaptive algorithm that enables a foraging system to approximate MSY *while* harvesting, by *implicitly* pursuing the optimal harvest rate.

Figure 1: A fishing robot. To balance sustainability and productivity, the robot must forage at an optimal rate called Maximum Sustainable Yield. [Artwork ©Christine Larson]

The foraging algorithm also must take noise into consideration. The robots' observation, actuation, and natural resource growth rate could all be noisy. We introduce these noises into the system and study their consequences. We show that the system sustainability is sensitive to a particular type of noise when the model is unknown. At last, the system might not have enough robots at the beginning. A foraging company will sell harvested resources, buy more robots, and harvest more. We simplify this businesses cycle by assuming that collected resources can be used directly to construct new robots, hence we introduce the robot construction problem.

2 Related Work

Lerman et al. study dynamic task allocation using multi-robot foraging system [2006]. They define a mathematical frame-

work for the foraging domain, including important concepts such as task, environment model, communication, observation and uncertainty. A formal analysis technique is provided to evaluate task performances. We follow up their framework and methodology closely.

People are particularly interested in the case where number of agents participate in foraging is very large, namely the swarm robot system. Examples in the nature are ant and bee colonies. Swarm robot foraging system has been studied by [Alers *et al.*, 2014; Ducatelle *et al.*, 2010; Hoff *et al.*, 2013; Hoff III, 2011; Liu *et al.*, 2007; Pini *et al.*, 2013], from the perspectives of task assignment, cooperation, decentralized control, energy conservation, etc.

An important aspect of a foraging task is the resource growth model, especially the logistic growth model [Verhulst, 1838]. Although the actual observed population could oscillate around predicted value, this model is still a powerful tool to study population growth [Cook, 1965]. Hjort et al. use logistic model to study fishery [1933]. They proposed an ideal population size and optimal harvest rate solution which is widely known as the Maximum Sustainable Yield (MSY) [Hjort *et al.*, 1933]. Resource population under logistic model is very sensitive to harvesting activities [Clark, 1973; Smith and Punt, 2001], hence maintaining MSY is challenging.

Our work extends [Song and Vaughan, 2013; 2012] that introduce sustainable robot foraging problem with logistic resource growth. Their work is the first to apply the concept of MSY to robot foraging. However, they assume the system is deterministic and robots have complete knowledge of the environment. They also assume that enough robots at the beginning and focused on the task assignment and scheduling. Their idea of reducing excess harvesting capacity by making some robots go to sleep [Song and Vaughan, 2013; 2012] is a critical component in our approach. A recent work of [Liemhetcharat *et al.*, 2015] follows closely to their work but emphasizes the productivity instead of sustainability with heterogeneous agents.

3 Problem Definition and Model Analysis

3.1 Environment

Logistic Model The resource population growth can be modeled by the Malthusian Model [Malthus, 1872] that assumes no upper bound on population. However, natural resource population does not grow without constraints due to the limited space and supply. The population grows asymptotically to an upper bound, namely the *carrying capacity* of an ecosystem. Let $P(t)$ denote the population at time t, $P(0)$ denote the initial resource population, and $\frac{dP(t)}{dt}$ denote the resource growth rate. The logistic model [Verhulst, 1838] is:

$$\frac{dP(t)}{dt} = kP(t) - bP^2(t) \tag{1}$$

$$P(t) = \frac{kP_0}{(k - bP_0)}e^{-kt} + bP_0 \tag{2}$$

$$\lim_{t\to\infty} P(t) = \frac{k}{b} \tag{3}$$

The constant b is generally very small comparing to k. However, as population grows, $bP^2(t)$ will have the effect of slowing the growth down. The upper bound P_∞ of population is $\frac{k}{b}$ as the infinite limit of $P(t)$. In addition to the ecosystem's carrying capacity (upper bound), we also address a sustainability issue (lower bound), denoted by P_0, known as the minimum viable population [Shaffer, 1981]. If resource population drops below P_0, the population will decline and extinct.

In order to simulate the system in a computer program, a differentiable equation is used to discretize Equation 1:

$$\Delta P(t) = kP(t) - bP^2(t) \tag{4}$$

where t is the discrete time step.

Harvesting Harvesting with logistic model requires keeping resource sustainable while maximizing productivity. The highest harvest rate should be confined by the maximum growth rate of the logistic model. The maximum value of $\Delta P(t)$ and its corresponding population size $P(t)$ can be evaluated by taking derivative of Equation 4 and set $\Delta P(t)' = 0$:

$$\Delta P^* = \frac{k^2}{4b};\ P^* = \frac{k}{2b} \tag{5}$$

The possible scenarios in harvesting are:

a. **Overharvesting**: when the harvest rate is too high, population will drop below P_0, and resource will be exploited.

b. **Underharvesting**: there are two cases of underharvesting. Either we harvest and maintain population size to be smaller than P^* or larger than P^*. In both cases, the resource growth rates are less than ΔP^* and thereby not productive.

c. **Maximum Sustainable Yield (MSY)**: the optimal solution requires the foragers to adjust their harvesting rate to make population reach the ideal size P^*, and then harvest at maximum growth rate ΔP^*. This optimal state is named MSY [Hjort *et al.*, 1933].

Here we also introduce the foraging settings and notations. The environment is defined as a set of resource *patches*, denoted by $\mathcal{A} = \{A_1, A_2, \ldots\}$. Each patch is a different type of resource. A single unit of resource is called a *puck*. The number of pucks grow according to the logistic model at every time interval s. The maximum increment of pucks ΔP^* is $\frac{k^2}{4b}$ during s, and $\frac{k^2}{4bs}$ for a unit time.

3.2 Information Availability

To achieve MSY, one needs to calculate P^* and ΔP^*, hence requires k and b. When k and b are provided, we say the system is *informed* and call its control algorithm an *informed algorithm*. Otherwise, the system needs to infer the relevant information and be *adaptive* with an *adaptive algorithm*.

The key observation here is that it is unnecessary to fit k and b directly as in [Oliver, 1964] to achieve MSY. The robots only need to know when resource population is near P^*. Recall P^* is associated with the maximum growth rate ΔP^*. Since the growth rate can be observed by the foraging robots, the task becomes detecting the maximum value of $\Delta P(t)$. The unique behavior of the logistic model is that, starting from P^*, as $P(t)$ decreases, $\Delta P(t)$ decreases; as $P(t)$ increases, $\Delta P(t)$ also decreases. We assume that the resource population is at the equilibrium (upper bound) or

close at the beginning. As robots start to forage, $P(t)$ will decrease monotonically; $\Delta P(t)$ will increase monotonically until ΔP^* is reached, and decrease afterwards. Therefore, the *stop condition* that indicates a patch reaches MSY can be detected when both $P(t)$ and $\Delta P(t)$ decrease, which we refer as the **simultaneous decrease** event. This phenomenon makes it possible to stop increasing harvesting power without completely knowing the logistic model. Later we will show how to detect such an event in practice.

3.3 Multi-Robot Foraging System

The foraging system includes the following components:

- Home: a place where collected resources are stored.
- A construction robot: stays at home and assembles new foraging robots with collected resource pucks. The construction time is denoted $t_{construct}$. It stores relevant information for the patches and other robots.
- Foraging robots: a team of robots is denoted $\mathcal{R} = \{R_1, R_2, \ldots\}$, where R_i denotes the set of robots that are assigned to patch A_i. A foraging robot carries one puck at a time. It takes time $t_{harvest}$ to harvest a puck, t_{unload} to unload the puck at home. The single trip time d_i for patch A_i is the Euclidean distance between home and A_i divided by robot velocity.

Assume at the beginning there are only a few robots. To achieve MSY, we need enough robots to harvest each resource patch until its population reaches P^*, and then these robots should harvest at maximum growth rate ΔP^*. Therefore, the tasks for the construction robot are constructing new robots, assigning them to the patches, and determining when to stop constructing. The tasks for the foraging robots are harvesting and sleep. They sleep because we might end up with slightly more robots than MSY requires. The overharvesting problem can be resolved by making robots sleep for t_{sleep} at home. The foraging robots assigned to the same patch need to consistently collect relevant information, communicate with each other, and adjust sleep time to approximate MSY. We do not simply deactivate/activate extra robots to ensure a fair distribution of work and rest time among robots.

4 Algorithms

Our goal is to design control algorithms for an autonomous, multi-robot foraging system, in the described setting. We present the informed and adaptive algorithms for the construction robot and foraging robots. We use finite state machines (FSM) to control robot behaviors.

4.1 Informed Algorithm

Informed Algorithm for Construction Robot

Let n_i^* be the optimal number of robots required by a patch i. Since a robot carries one puck at a time, the harvest rate should be the inverse of its round trip time, which is given by $rtt_i = t_{harvest} + 2d_i + t_{unload} + t_{sleep}$. We then establish the following equation:

$$\frac{n_i^*}{rtt_i} = \frac{k^2}{4bs} \tag{6}$$

Algorithm 1 Construction Robot Algorithm (Informed)

Require: information of the construction robot, foraging robots and patches at time step t
  return behavior of the construction robot at time $t+1$
  for $i := 1 \to |\mathcal{A}|$ **do**
    if $|R_i| \geqslant n_i^*$ **then**
      $e_i \leftarrow$ **true**
    end if
  end for
  for $i := 1 \to |\mathcal{A}|$ **do**
    $v_i \leftarrow u_i / w_i$
  end for
  if not all entries in $\mathbf{e}$ are **false and** $state = idle$ **and** $\min(\mathbf{v}) \geqslant 1$ **then**
    $state \leftarrow construct$, $timer \leftarrow t_{construct}$
    $\mathbf{u} \leftarrow \mathbf{u} - \mathbf{w}$
  end if
  if $state = construct$ **then**
    if $timer > 0$ **then**
      $timer \leftarrow timer - 1$
    else
      a new robot is ready to go
      $i^* \leftarrow \arg\min_{i \in [|\mathcal{A}|]} \{v_i \mid e_i = \textbf{false}\}$
      assign the new robot to patch i^*, append robot to R_i
      $state \leftarrow idle$
    end if
  end if

Ideally, the sleep time of a foraging robot should be zero. Given that the logistic parameters k and b are known, we have

$$n_i^* = (t_{harvest} + 2d_i + t_{unload})\frac{k^2}{4bs} \tag{7}$$

The algorithm for an informed construction robot is shown in Algorithm 1. The variables are defined below:

- $state \in \{construct, idle\}$ indicates the state of a construction robot;
- $\mathbf{e} \in \{true, false\}^{|\mathcal{A}|}$, a boolean vector for the construction robot, initialized to be $false$ for all resource patches. e_i denotes whether patch A_i has enough robots;
- $\mathbf{u} \in \mathbb{Z}_+^{|\mathcal{A}|}$) records the amount of pucks at home, where u_i is for the ith type of resource;
- $\mathbf{w} \in \mathbb{Z}_+^{|\mathcal{A}|}$) indicates the amount of pucks needed to construct a robot, where w_i is for the ith type of resource;
- $timer$ indicates the remaining time of current state.

At each time step, the construction robot decides whether each patch has enough robots. If so, the construction robot stops constructing robot for that patch. If not, and if there are enough pucks to construct a robot at home, it starts to assemble a new robot. Upon finishing, the new robot is assigned to the patch with the scarcest resource in terms of demand to construct a new robot. This can be calculated by finding the minimum element in a vector $\mathbf{v}$, which is the proportion of the amount of pucks in storage and the amount required to construct a new robot.

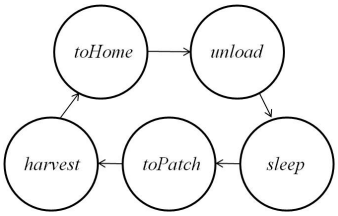
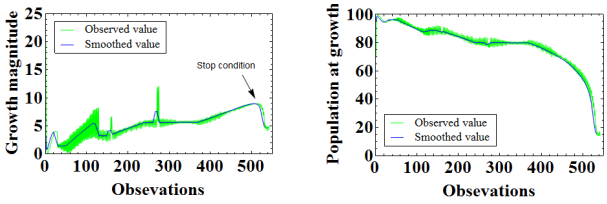


Figure 2: Foraging robot finite state machine



(a) Observed growth (b) Population size

Figure 3: Determining stop condition.

Informed Algorithm for Foraging Robots

The algorithm is shown as a state machine in Figure 2. The states are $toHome$, $unload$, $sleep$, $toPatch$, and $harvest$. The state transition is based on a $timer$ variable, which counts the remaining time for the current state. A robot will always harvest its assigned patch. When a robot transits from $toPatch$ state to $harvest$, it measures the amount of pucks $P(t)$ to prevent destroying the resource. If $P(t)$ is less than or equal to the logistic lower bound P_0, the robot will stop harvesting.

It is critical that the solution of n_i^* from Equation 7 may not be an integer. A patch A_i will have slightly more robots than the optimal solution n_i^*. To resolve the overharvesting problem, when a robot arrives home, it updates sleep time and go to sleep. Ideally, the average amount of pucks in any patch must be close to P^*. We calculate the increment of the sleep time Δt_{sleep} for the moment t as follows [Song and Vaughan, 2013; 2012]:

$$P_{error}(t) = P^* - P(t) \tag{8}$$

$$\Delta t_{sleep} = \begin{cases} K_{in} P_{error}(t) & P_{error}(t) > 0 \\ K_{de} P_{error}(t) & P_{error}(t) < 0 \\ 0 & P_{error}(t) = 0 \end{cases} \tag{9}$$

K_{in} and K_{de} are constants that represent increase and decrease rate. When a patch is underharvested ($e_i = false$), its assigned robots should not sleep, and $t_{sleep} = 0$. Otherwise, t_{sleep} is updated by Δt_{sleep}:

$$t_{sleep} \leftarrow t_{sleep} + \Delta t_{sleep} \tag{10}$$

4.2 The Adaptive Algorithm

Adaptive Algorithm for Construction Robot

The algorithm for an adaptive construction robot is similar to the informed one, but it cannot directly calculate the optimal number of robots n_i^*. The construction robot relies on the foraging robots to collect information and decide when MSY is reached. It updates e_i to be true for patch A_i if a foraging robot in $\mathcal{R}_i$ sends a signal for it to stop, and stops constructing robot for that patch.

Adaptive Algorithm for Foraging Robots

In theory, the **simultaneous decrease** phenomenon signals the time to stop building new robots. However, the events of resource growth, robot harvesting, and robot observation are all discrete, while the latter two are not uniformly distributed over time. Although the macroscopic simultaneous decrease pattern will follow the ideal situation, the observed $P(t)$ and $\Delta P(t)$ can fluctuate, as shown in Figure 3. We assume that all foraging robots assigned to a patch can broadcast their recent collected patch information. Whenever a robot observes a population growth at a patch, it shares the magnitude of growth and the population size before the growth. The green curve in Figure 3a shows observed population growth magnitude for one patch, and the green curve in Figure 3b shows corresponding population size. To detect the simultaneous decrease, several 1D filters for signal processing can be used. We use a Gaussian filter of $\sigma = 10$ with kernel size 20 to smooth the observed data.

Therefore, the algorithm for the adaptive foraging robot is different from the informed one. The behavior when a robot arrives at the patch is shown in Algorithm 2. All foraging robots assigned to the same patch share collected information via the following five queues:

- Q records the observed population size.
- Q_α records the growth magnitude, which is the green curve in Figure 3a. If a robot detects its current observed population $P(t)$ is greater than the last entry of Q, the robot concludes that it observes a population growth. It records the growth magnitude $P(t) - Q.back()$, and enqueues the value into Q_α.
- $\tilde{Q}_\alpha$ records the smoothed growth magnitude, which is the blue curve in Figure 3a. Whenever Q_α is updated, a new smoothed value is calculated and enqueued.
- Q_β records the population size before growth, which is the green curve in Figure 3b. Whenever Q_α is updated, $Q.back()$ is enqueued into this queue.
- $\tilde{Q}_\beta$ records a smoothed value of entries in Q_β, which is the blue curve in Figure 3b. Whenever Q_β is updated, a new smoothed value is calculated and enqueued.

The foraging robots can use $\tilde{Q}_\alpha$ and $\tilde{Q}_\beta$ to detect the simultaneous decrease at its patch, and send a signal to construction robot indicating that the stop condition is reached. However, such detection is delayed due to the filtering window.

Comparing to the informed algorithm, Equation 11 is used instead of 8 to update sleep time. Now $P_{error}(t)$ is calculated using the population size $\hat{P}$ before the simultaneous decrease:

$$P_{error}(t) = \hat{P} - P(t) \tag{11}$$

Algorithm 2 Foraging Robot Algorithm (Adaptive)

if $state = toPatch$ **and** $timer = 0$ **then**
  $state \leftarrow harvest$
  observe current resource amount $P(t)$
  if $P(t) > Q.back()$ **then**
    update Q_α,$\tilde{Q}_\alpha$,Q_β,$\tilde{Q}_\beta$ in order
    if $\tilde{Q}_\alpha[-1] < \tilde{Q}_\alpha[-2]$ **and** $\tilde{Q}_\beta[-1] < \tilde{Q}_\beta[-2]$ **then**
      send a signal to construction robot, $e_i \leftarrow$ **true**
      $\hat{P} \leftarrow Q.back()$
    end if
  end if
  $Q.enqueue(P(t))$
  if $P(t) > P_0$ **then**
    $timer \leftarrow t_{harvest}$, $carryPuck \leftarrow$ **true**
    $P(t) \leftarrow P(t) - 1$
  end if
end if

5 Experiments and Results

5.1 Experiment Setup

We compare the performance of the informed algorithm and the adaptive algorithm. We implement a simulator using Player/Stage [Gerkey *et al.*, 2003]. The parameters are chosen to be consistent with [Song and Vaughan, 2013].

- Resources: the number of patches $|\mathcal{A}| = 3$. We use the same logistic model parameters for all patches, $k = .4$, $b = .004$. The choice of k and b do not affect our main conclusions. The base is located at (.5, .5). The patches are located at (.1, .1), (.2, .8), and (.7, .7), so they require different number of robots to harvest. The lower bound, upper bound, and initial population are $P_0 = 15$, $P_\infty = 100$, $P(0) = 50$, respectively. The resource growth time interval $s = 100$. The MSY is reached when $P^* = 50$, $\Delta P^* = .1$ per unit time, or equivalently, $\Delta P^* = 10$ per s time steps.
- Construction robot: the construction time for a new robot $t_{construct} = 1200$. The amount of pucks needed to construct a new robot $\mathbf{w} = [3, 2, 1]$.
- Foraging robots: $t_{harvest} = 10$, $t_{unload} = 8$, $velocity = .004$. The initial number of robots assigned to each patch is 1.

Based on Equation 7, the optimal numbers of robots for the patches are $n_1^* = 30.08$, $n_2^* = 23.01$, $n_3^* = 15.94$. Thus, the optimal solution allocates $31, 24, 16$ robots to each patch. The parameters we use for our algorithms are $K_{in} = 3.0$, $K_{de} = 2.0$ for the informed sleep time adjustment. $K_{in} = 3.0$, $K_{de} = .2$ for the adaptive sleep time adjustment. Note K_{de} for the adaptive algorithm is small, that is, robots decrease their sleep time slowly. We want to be conservative to prevent overharvesting in the adaptive setting.

5.2 Results

The total length of experiment is $20,0000$ time steps and we record data every 200 steps. The goal is to show the correctness of the informed algorithm, and that the adaptive algorithm performs closely to the informed one under incomplete information. Figure 4a and 4b show the resource population over time. Both algorithms converge near $P^* = 50$, while no patch's population drop near P_0, hence sustainability is maintained. With $P^* = 50$, the patch growth rate converges to $\Delta P^* = 10$. Figure 4c and 4d confirm that the foraging robots eventually deliver pucks at a rate that is on average equal to the maximum patch growth rate, thus productivity is meet.

We observe more fluctuations in the adaptive algorithm. This phenomenon is expected since the informed algorithm knows the model parameters and stop constructing robot immediately when optimal number of robot is reached. Figure 4e illustrates the correctness of the informed algorithm, where exactly $31, 24, 16$ robots are constructed. The adaptive algorithm constructs $34, 26, 19$ robots, which are slightly more than desired due to the delayed detection of stop condition. The extra robots cause excess harvesting capacity. Therefore, in Figure 4b, before convergence, the population drops below 50 due to overharvesting. Then the algorithm detects this problem, stops constructing robots, and foraging robots start to sleep, overharvesting is reduced and results gradually converge to MSY. But since more robots are constructed in the adaptive algorithm, it is more difficult to adjust sleep time to maintain MSY; thereby we observe more fluctuations.

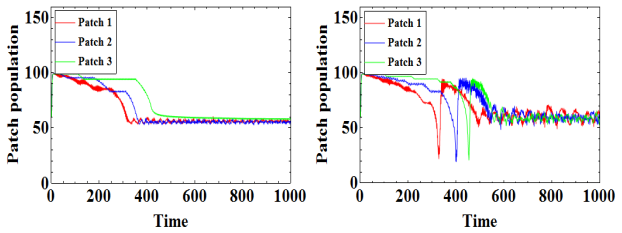
The productivity results are shown in Figure 4f, where the total number of collected pucks is counted. The adaptive algorithm performs comparable to the informed algorithm.

5.3 Effects of Noise

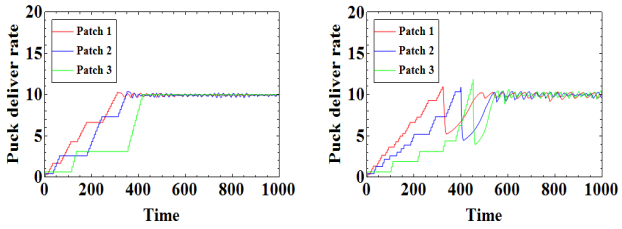
We further introduce noises into resource growth, robot actuation, and observation. The noises are Gaussian random variables with mean zero and standard deviations equal 10% of the affected variable (Equation 12). The patch population now does not completely grow according to the logistic model since noise is added to the growth amount. For actuation, noise is added to variables $t_{harvest}, t_{unload}, t_{sleep}$, and the trip time d_i. Noise is also added to the robot's observation of population size.

$$x \leftarrow x + \mathcal{N}(0, (0.1 * x)^2) \tag{12}$$

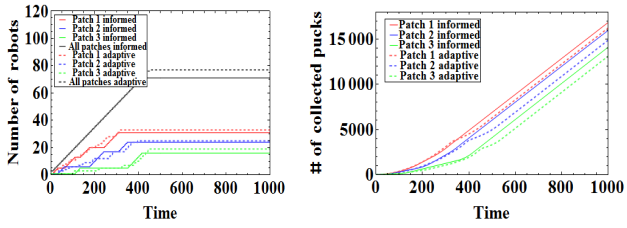
Results for the informed algorithm are shown in Figure 5a and 5b. The informed algorithm is robust to the noise, although the performance fluctuates more comparing to Figure 4a and 4c. An important finding is that the adaptive algorithm is sensitive to noise. Figure 5c and 5d shows an example of failure to reach MSY, where the red patch is overharvested and the population decreases to the minimum viable population, and the other two patches are underharvested. By performing a grid search over different combinations and sizes of the noises, the adaptive algorithm is robust to noise in the actuation and population growth. Problems arise from noise added to the robots' observations of population size. Recall that the observed population size determines the stop condition. The incorrectly observed population size makes detecting stop condition very difficult. By a grid search, the algorithm can only accept a Gaussian noise with a standard deviation equals 1.5% of the observed population size with the presence of other noises, as shown in Figure 5e and 5f. Results suggest that, under incomplete model information, with



(a) Population (informed) (b) Population (adaptive)

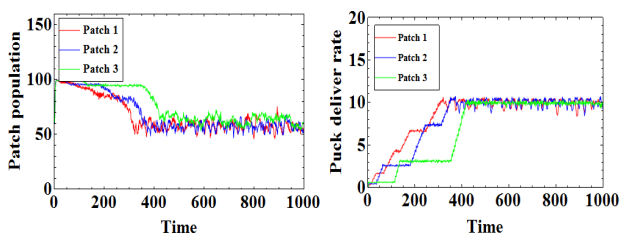


(c) Delivery rate (informed) (d) Delivery rate (adaptive)

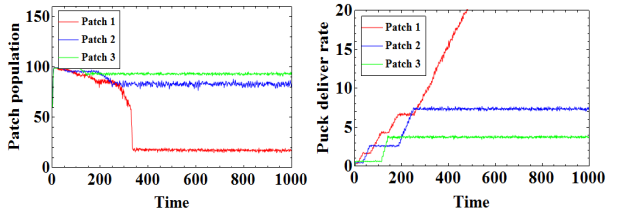


(e) Number of robots (f) Performance

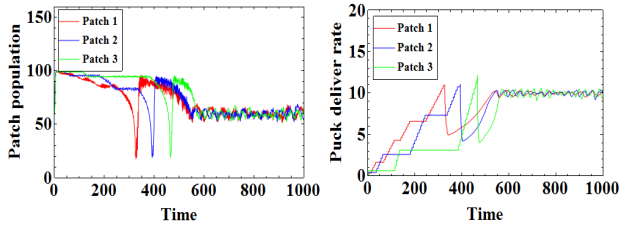
Figure 4: The informed vs. the adaptive algorithm in terms of patch population, puck delivery rate, number of robots constructed, and total number of pucks collected.



(a) Population (informed) (b) Delivery rate (informed)



(c) Population (adaptive) (d) Delivery rate (adaptive)



(e) Population (adaptive) (f) Delivery rate (adaptive)

Figure 5: Top row: the effect of 10% Gaussian noise on the informed algorithm. Middle row: the effect of 10% Gaussian noise on the adaptive algorithm. Bottom row: the effect of 1.5% Gaussian noise on the adaptive algorithm

randomness in population growth, the accurate observation is a key factor for the foraging system's decision. It suggests that monitoring the population size closely and accurately is necessary to achieve MSY.

6 Conclusion and Future Work

The foraging industry today increasingly relies on automatic operations. Biology and environmental studies communities have presented models and rules to maintain sustainability. However, the robots present new challenges: they are efficient but potentially damaging. They require adaptive algorithms to respond to uncertainties in the environment. In this work, we extended the previous work of multi-robot and multi-patch foraging system while resource growth model is logistic. We have the following two main contributions.

First, we overcome the practical issue of unknown model parameters for MSY foraging problem by developing an adaptive algorithm which detects the simultaneous decrease event. An alternative approach is to set industrial foraging regulations. But they are often global (e.g., fishing off-seasons for the entire bay area) and economically inefficient since each area is unique. Our patch growth model captures the individual differences. Manually estimating individual growth models is expensive, and we show that the robots can adapt on-line.

Second, we introduce multiple sources of noise into the system and show that the foraging system could still achieve MSY, but require accurate observation of population. The implication is that the foraging algorithm is robust to noise in robot actuation and resource growth. However, foraging industry needs to closely monitor the population change to maintain sustainability.

For future work, a robot failure, repair, and replacement pipeline can be introduced, which requires the algorithms to be robust to resulted sudden harvest rate changes. Also, in the adaptive algorithm, we require foraging robots to broadcast collected information and send signal home. Instead of broadcasting, the foraging robots can exchange information when they meet on their way, as inspired by the communication behaviors of the insects. In addition, we plan to adopt a more natural resource growth model. For example, the population growth rate often has seasonal fluctuations and requires foraging robots to detect and adapt to this phenomenon.

References

[Alers *et al.*, 2014] Sjriek Alers, Karl Tuyls, Bijan Ranjbar-Sahraei, Daniel Claes, and Gerhard Weiss. Insect-inspired robot coordination: Foraging and coverage. In *ALIFE*, pages 761–768, 2014.

[Clark, 1973] Colin W Clark. The economics of overexploitation. *Science*, 181(4100):630–634, 1973.

[Cook, 1965] LM Cook. Oscillation in the simple logistic growth model. Nature Publishing Group, 1965.

[Ducatelle *et al.*, 2010] Frederick Ducatelle, Gianni A Di Caro, and Luca M Gambardella. Cooperative self-organization in a heterogeneous swarm robotic system. In *GECCO*, pages 87–94, 2010.

[Gerkey *et al.*, 2003] Brian Gerkey, Richard T Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323, 2003.

[Hjort *et al.*, 1933] Johan Hjort, Gunnar Jahn, and Per Ottestad. *The optimum catch*. I kommisjon hos J. Dybwad, 1933.

[Hoff *et al.*, 2013] Nicholas Hoff, Robert Wood, and Radhika Nagpal. Distributed colony-level algorithm switching for robot swarm foraging. In *DARS*, pages 417–430. 2013.

[Hoff III, 2011] Nicholas Roosevelt Hoff III. *Multi–Robot Foraging for Swarms of Simple Robots*. PhD thesis, Citeseer, 2011.

[Lerman *et al.*, 2006] Kristina Lerman, Chris Jones, Aram Galstyan, and Maja J Matarić. Analysis of dynamic task allocation in multi-robot systems. *In IJRR*, 25(3):225–241, 2006.

[Liemhetcharat *et al.*, 2015] Somchaya Liemhetcharat, Rui Yan, and Keng Peng Tee. Continuous foraging and information gathering in a multi-agent team. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1325–1333. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[Liu *et al.*, 2007] Wenguo Liu, Alan Winfield, Jin Sa, Jie Chen, and Lihua Dou. Strategies for energy optimisation in a swarm of foraging robots. In *Swarm robotics*, pages 14–26. Springer, 2007.

[Malthus, 1872] Thomas Robert Malthus. *An Essay on the Principle of Population Or a View of Its Past and Present Effects on Human Happiness, an Inquiry Into Our Prospects Respecting the Future Removal Or Mitigation of the Evils which it Occasions by Rev. TR Malthus*. Reeves and Turner, 1872.

[Oliver, 1964] FR Oliver. Methods of estimating the logistic growth function. *Applied statistics*, pages 57–66, 1964.

[Pini *et al.*, 2013] Giovanni Pini, Arne Brutschy, Carlo Pinciroli, Marco Dorigo, and Mauro Birattari. Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adaptive behavior*, 21(2):118–136, 2013.

[Shaffer, 1981] Mark L Shaffer. Minimum population sizes for species conservation. *BioScience*, 31(2):131–134, 1981.

[Smith and Punt, 2001] T Smith and AE Punt. The gospel of maximum sustainable yield in fisheries management: birth, crucifixion and reincarnation. *Conservation of exploited species. Edited by JD Reynolds, GM Mace, KH Redford, and JG Robinson. Cambridge University Press, Cambridge, UK*, pages 41–66, 2001.

[Song and Vaughan, 2012] Zhao Song and Richard T Vaughan. Multi-robot, multi-patch foraging with maximum sustainable yield. In *ALife XIII*, 2012.

[Song and Vaughan, 2013] Zhao Song and Richard T Vaughan. Sustainable robot foraging: adaptive fine-grained multi-robot task allocation for maximum sustainable yield of biological resources. In *IROS*, pages 3309–3316, 2013.

[Verhulst, 1838] P.F. Verhulst. Notice sur la loi que la population suit dans son acroissement. *Corr. Math. et Phys*, 10:113, 1838.

[Winfield, 2009] Alan FT Winfield. Foraging robots. In *Encyclopedia of Complexity and Systems Science*, pages 3682–3700. Springer, 2009.