Analysis of Dynamic Task Allocation in Multi-Robot Systems

Kristina Lerman[1], Chris Jones[2], Aram Galstyan[1] and Maja J Matarić[2]

1. Information Sciences Institute
2. Computer Science Department
University of Southern California, Los Angeles, CA 90089-0781, USA
{lerman|galstyan}@isi.edu, {cvjones|maja}@robotics.usc.edu

Abstract

Dynamic task allocation is an essential requirement for multi-robot systems functioning in unknown dynamic environments. It allows robots to change their behavior in response to environmental changes or actions of other robots in order to improve overall system performance. Emergent coordination algorithms for task allocation that use only local sensing and no direct communication between robots are attractive because they are robust and scalable. However, a lack of formal analysis tools makes emergent coordination algorithms difficult to design. In this paper we present a mathematical model of a general dynamic task allocation mechanism. Robots using this mechanism have to choose between two types of task, and the goal is to achieve a desired task division in the absence of explicit communication and global knowledge. Robots estimate the state of the environment from repeated local observations and decide which task to choose based on these observations. We model the robots and observations as stochastic processes and study the dynamics of individual robots and the collective behavior. We analyze the effect that the number of observations and the choice of decision functions have on the performance of the system. We validate the mathematical models on a multi-foraging scenario in a multi-robot system. We find that the model's predictions agree very closely with experimental results from sensor-based simulations.

1 Introduction

Twenty years ago it was considered ground-breaking for a mobile robot to move around an unstructured environment at reasonable speeds. In the years since, advancements in both hardware mechanisms and software architectures and algorithms have resulted in quite capable mobile robot systems. Provided with this baseline competency of individual robots, increasing attention has been paid to the study of Multi-Robot Systems (MRS), and in particular *distributed* MRS with which the remainder of this paper is concerned. In a distributed MRS there is no centralized control mechanism – instead, each robot operates independently under local sensing and control, with coordinated system-level behavior arising from local interactions among the robots and between the robots and the task environment. The effective design of coordinated MRS is restricted by the lack of formal design tools and methodologies. The design of single robot systems

(SRS) has greatly benefited from the formalisms provided by control theory – the design of MRS is in need of analogous formalisms.

For a group of robots to effectively perform a given system-level task, the designer must address the question of which robot should do which task and when [4]. The process of assigning individual robots to sub-tasks of a given system-level task is called *task allocation*, and it is a key functionality required of any MRS. *Dynamic* task allocation is a class of task allocation in which the assignment of robots to sub-tasks is a dynamic process and may need to be continuously adjusted in response to changes in the task environment or group performance. The problem of task allocation in a *distributed* MRS is further compounded by the fact that task allocation must occur as a result of a distributed process as there is no central coordinator available to make task assignments. This increases the problem's complexity because, due to the local sensing of each robot, no robot has a complete view of the world state. Given this incomplete and often noisy information, each robot must make local control decisions about which actions to perform and when, without complete knowledge of what other robots have done in the past, are doing now, or will do in the future.

There are a number of task allocation models and philosophies. Historically, the most popular approaches rely on intentional coordination to achieve task allocation [21]. In those, the robots coordinate their respective actions explicitly through deliberate communications and negotiations. Due to scaling issues, such approaches are primarily used in MRS consisting of a relatively small number of robots (i.e., less than 10). Task allocation through intentional coordination remains the preferred approach because it is better understood, easier to design and implement, and more amenable to formal analysis [4].

As the size of the MRS grows, the complexity of the design of intentional approaches increases due to increased demands in communication bandwidth and computational abilities of individual robots. Furthermore, complexity introduced by increased robot interactions makes such systems much more difficult to analyze and design. This leads to the alternative to intentional coordination, namely, task allocation through utilizing emergent coordination. In systems using emergent coordination, individual robots coordinate their actions based solely on local sensing information and local interactions. Typically, there is very little or no direct communication or explicit negotiations between robots. They are, therefore, more scalable to larger numbers of robots and are more able to take advantage of the robustness and parallelism provided by the aggregation of large numbers of coordinated robots. The drawback of task allocation as achieved through emergent coordination mechanisms is that such systems can be difficult to design, solutions are commonly sub-optimal, and since coordination is achieved through many simultaneous local interactions between various subsets of robots, predictive analysis of expected system performance is difficult.

As MRS composed of ever-larger numbers of robots become available, the need for task allocation through emergent coordination will increase. To address the lack of formalisms in the design of such MRS, in this article we present and experimentally verify a predictive mathematical model of dynamic task allocation for MRS using emergent coordination. Such a formal model of task allocation is a positive step in the direction of placing the design of MRS on a formal footing.

In Section 3 we describe a general mechanism for task allocation in dynamic environments. This is a distributed mechanism based on local sensing. In Section 4 we present a mathematical model of the collective behavior of an MRS using this mechanism and study its performance under a variety of conditions. We validate the model in a multi-foraging domain. In Section 5 we define the experimental task domain of multi-foraging, robot controllers and the simulation environment. Finally, in Section 6 we compare the predictions of mathematical models with the results from sensor-based simulations. We conclude the paper with a discussion of the approach and the results.

2 Related Work

Mathematical modeling and analysis of the collective behavior of MRS is a relatively new field with approaches and methodologies borrowed from other fields, including mathematics, physics, and biology. Recently, a number of researchers attempted to mathematically analyze multi-robot systems by using phenomenological models of the type present here. Sugawara *et al.* [23, 24] developed a simple model of cooperative foraging in groups of communicating and non-communicating robots. Kazadi *et al.* [11] studied the general properties of multi-robot aggregation using phenomenological macroscopic models. Agassounon and Martinoli [1] presented a model of aggregation in which the number of robots taking part in the clustering task is based on the division of labor mechanism in ants. These models are *ad-hoc* and domain specific, and the authors give no explanation as to how to apply such models to other domain. In earlier works we have developed a general framework for creating phenomenological models of collective behavior in groups of robots [16, 18]. We applied this framework to study collaborative stick-pulling in a group of reactive robots [17] and foraging in robots [13].

Most of the approaches listed above are implicitly or explicitly based on stochastic processes theory. Another example of the stochastic approach is the probabilistic microscopic model developed by Martinoli and coworkers [19, 20, 8] to study collective behavior of a group of robots. Rather than compute the exact trajectories and sensory information of individual robots, Martinoli *et al.* model each robot's interactions with other robots and the environment as a series of stochastic events, with probabilities determined by simple geometric considerations. Running several series of stochastic events in parallel, one for each robot, allowed them to study the group behavior of the multi-robot system.

So far very little work has been done on mathematical analysis of multi-robot systems in dynamic environments. We have recently extended [14] the stochastic processes framework developed in earlier work to robots that change their behavior based on history of local observations of the (possibly changing) environment [15]. In the current paper we develop these ideas further, and present the exact stochastic model of the system, in addition to the phenomenological model.

Closest to ours is the work of Huberman and Hogg [7], who mathematically studied collective behavior of a system of adaptive agents using game dynamics as a mechanism for adaptation. In game dynamical systems, winning strategies are rewarded, and agents use the best performing strategies to decide their next move. Although their adaptation mechanism is different from our dynamic task allocation mechanism, their analytic approach is similar to ours, in that it is based on the theory of stochastic processes. Others have mathematically studied collective behavior of systems composed of large numbers of concurrent learners [25, 22]. These are microscopic models, which only allow one to study collective behavior of relatively small systems of a few robots. We are interested in macroscopic approaches that enable us to directly study collective behavior in large systems. Our work differs from earlier ones in another important way: we systematically compare theoretical predictions of mathematical models with results of experiments carried out in a sensor-based simulator.

3 Dynamic Task Allocation Mechanism

The dynamic task allocation scenario we study considers a world populated with tasks of T different types and robots that are equally capable of performing each task but can only be assigned to one type at any given time. For example, the tasks could be targets of different priority that have to be tracked, different types of explosives that need to be located, etc. Additionally, a robot cannot be idle — each robot is always performing a task at any given time. We introduce the notion of a robot state as a shorthand for the type of task the robot is assigned to service. A robot may switch its state according to its control policy when it determines it is appropriate to do so. However, needlessly switching tasks is to be avoided, since in physical robot systems, this can involve complex physical movement that requires time to perform.

The purpose of task allocation is to assign robots to tasks in a way that will enhance the performance of the system, which typically means reducing the overall execution time. Thus, if all tasks take an equal amount of time to complete, in the best allocation, the fraction of robots in state i will be equal to the fraction of tasks of type i. In general, however, the desired allocation could take other forms — for example, it could be related to the relative reward or cost of completing each task type — without change to our approach. In the dynamic task allocation scenario, the number of tasks and the number of available robots are allowed to change over time, for example, by adding new tasks, deploying new robots, or removing malfunctioning robots.

The challenge faced by the designer is to devise a mechanism that will lead to a desired task allocation in a distributed MRS even as the environment changes. The challenge is made even more difficult by the fact that robots have limited sensing capabilities, do not directly communicate with other robots, and therefore, cannot acquire global information about the state of the world, the initial or current number of tasks (total or by type), or the initial or current number of robots (total or by assigned type). Instead, robots can sample the world (assumed to be finite) — for example, by moving around and making local observations of the environment. We assume that robots are able to observe tasks and discriminate their types. They may also be able to observe and discriminate the task states of other robots.

One way to give the robot an ability to respond to environmental changes (including actions of other robots) is to give a robot an internal state where it can store its knowledge of the environment as captured by its observations [9, 14]. The observations are stored in a rolling history window of finite length, with new observations replacing the oldest ones. The robot consults these observations periodically and updates its task state according to some transition function specified by the designer. In an earlier work we showed [9, 15] that this simple dynamic task allocation mechanism leads to the desired task allocation in a multi-foraging scenario.

In the following sections we present a mathematical model of dynamic task allocation and study the role that transition function and the number of observations (history length) play in the performance of a multi-foraging MRS. In Section 4.1, we present a model of a simple scenario in which robots base their decisions to change state solely on observations of tasks in the environment. We study the simplest form of the transition function, in which the probability to change state to some type is proportional to the fraction of existing tasks of that type. In Section 6.1 we compare theoretical predictions with no adjustable parameters to experimental data and find excellent agreement. In Section 4.2 we examine the more complex scenario where the robots base their decisions to change task state on the observations of both existing task types and task states of other robots. In Section 6.2 we study the consequences of the choice of the transition function and history length on the system behavior and find good agreement with the experimental data.

4 Analysis of Dynamic Task Allocation

As proposed in the previous section, a robot may be able to adapt to a changing environment in the absence of complete global knowledge if it is able to make and remember local observations of the environment. In the treatment below we assume that there are two types of tasks — arbitrarily referred to as *Red* and *Green*. This simplification is for pedagogical reason only; the model can be extended to a greater number of task types.

During a sufficiently short time interval, each robot can be considered to belong to the *Green* or *Red* task state. This is a very high level, coarse-grained description. In reality, each state is composed of several robot actions and behaviors, for example, searching for new tasks, detecting and executing them, avoiding obstacles, *etc.* However, since we want the model to capture how the fraction of robots in each task state evolves in time, it is a sufficient level of abstraction to consider only these two states. If we find that additional levels of detail are required to explain system behavior, we can elaborate the model by breaking each of the high level states into its underlying components.

4.1 Observations of Tasks Only

In this section we study dynamic task allocation mechanism in which robots make decisions to switch task states based solely on observations of available tasks. Let m_r and m_g be the numbers of the observed *Red* and *Green* tasks, respectively, in a robot's memory or history window. The robot chooses to change its state, or the type of task it is assigned to execute, with probabilities given by transition functions $f_{g\to r}(m_r, m_g)$ (probability of switching to *Red* from *Green*) and $f_{r\to g}(m_r, m_g)$ (probability of switching to *Green* from *Red*). We would like to define transition rules so that the fraction of time the robot spends in the *Red* (*Green*) state be equal to the fraction of *Red* (*Green*) tasks. This will assure that on average the number of *Red* and *Green* robots reflect the desired task distribution. Clearly, if the robots have global knowledge about the numbers of *Red* and *Green* tasks M_r and M_g, then each robot could choose each state with probability equal to the fraction of the tasks of corresponding type. Such global knowledge is not available; hence, we want to investigate how incomplete knowledge of the environment (through local observations), as well as the dynamically changing environment (e.g., changing ratio of *Red* and *Green* tasks), affects task allocation.

4.1.1 Modelling Robot Observations

As explained above, the transition rate between task execution states depends on robot's observations stored in its history. In our model we assume that a robot makes an observation of a task with a time period τ. For simplicity, by an observation we mean here detecting a task, such as a target to be monitored, mine to be cleared or an object to be gathered. Therefore, observation history of length h comprises of the number of *Red* and *Green* tasks a robot has observed during a time interval $h\tau$. We assume that τ has unit length and drop it. The process of observing a task is given by a Poisson distribution with rate $\lambda = \alpha M^0$, where α is a constant characterizing the physical parameters of the robot such as its speed, view angles, etc., and M^0 is the number of tasks in the environment. This simplification is based on the idea that robot's interactions with other robots and the environment are independent of the robot's actual trajectory and are governed by probabilities determined by simple geometric considerations. This simplification has been shown to produce remarkably good agreements with experiments [20, 8].

Let $M_r(t)$ and $M_g(t)$ be the number of *Red* and *Green* tasks respectively (can be time dependent), and let $M(t) = M_r(t) + M_g(t)$ be the total number of tasks. The probability that in the time interval $[t-h, t]$ the robot has observed exactly m_r and m_g tasks is the product of two Poisson distributions:

$$P(m_r, m_g) = \frac{\lambda_r^{m_r} \lambda_g^{m^g}}{m_r! m_g!} e^{-\lambda_r - \lambda_g} \tag{1}$$

where λ_i , $i = r, g$, are the means of the respective distributions. If the task distribution does not change in time, $\lambda_i = \alpha M_i h$. For time dependent task distributions, $\lambda_i = \alpha \int_{t-h}^{t} dt' M_i(t')$.

4.1.2 Individual Dynamics: The Stochastic Master Equation

Let us consider a single robot that has to decide between executing *Red* and *Green* tasks in a closed arena and makes a transition to *Red* and *Green* states according to its observations. Let $p_r(t)$ be the probability that a robot is in the *Red* state at time t. The equation governing its evolution is

$$\frac{dp_r}{dt} = \varepsilon(1 - p_r) f_{g \to r} - \varepsilon p_r f_{r \to g} \tag{2}$$

where ε is the rate at which the robot makes decisions to switch its state, and $f_{g \to r}$ and $f_{r \to g}$ are the corresponding transitions probabilities between the states. As explained above, these probabilities depend on the robot's history — the number of tasks of either type it has observed during the time interval h preceding the transition. If the robots have global knowledge about the numbers of *Red* and *Green* tasks M_r and M_g, one could choose the transition probabilities as the fraction of tasks of corresponding type, $f_{g \to r} \propto M_r/(M_r + M_g)$ and $f_{r \to g} \propto M_g/(M_r + M_g)$. In the case when the global information is not available, it is natural to use similar transition probabilities using robots' local estimates:

$$\begin{aligned} f_{g \to r}(m_r, m_g) &= \frac{m_r}{m_r + m_g} \equiv \gamma_r(m_r, m_g) \\ f_{r \to g}(m_r, m_g) &= \frac{m_g}{m_r + m_g} \equiv \gamma_g(m_r, m_g) \end{aligned} \tag{3}$$

Note that $\gamma_r(m_r, m_g) + \gamma_g(m_r, m_g) = 1$ whenever $m_r + m_g > 0$, *e.g.*, whenever there is at least one observation in the history window. In the case when there are no observations in history, $m_r = m_g = 0$, robots will choose either state with probability 1/2 as it follows from taking the appropriate limits in Equation 4. Hence, we supplement Equation 4 with $f_{g \to r}(0,0) = f_{r \to g}(0,0) = 0$ to assure that robots do not change their state when the history window does not contain any observations.

Equation 2, together with the transition rates as shown in Equation 4, determines the evolution of the probability density of a robot's state. It is a stochastic equation since the coefficients (transition rates) depend on random variables m_r and m_g. Moreover, since the robot's history changes gradually, the values of the coefficients at different times are correlated, hence making the exact treatment very difficult. We propose, instead, to study the problem within the *annealed* approximation: we neglect time–correlation between robot's histories at different times, assuming instead that at any time the real history $\{m_r, m_g\}$ can be replaced by a random one drawn from the Poisson distribution Equation 1. Next, we average Equation 2 over all histories to obtain

$$\frac{dp_r}{dt} = \varepsilon\overline{\gamma}_r(1 - n_r) - \varepsilon\overline{\gamma}_g n_r \tag{4}$$

Here $\overline{\gamma}_r$ and $\overline{\gamma}_g$ are given by

$$\overline{\gamma}_r = \sum_{r,g} P(r,g)\frac{r}{r+g}, \overline{\gamma}_g = \sum_{r,g} P(r,g)\frac{g}{r+g} \tag{5}$$

where $P(m_r, m_g)$ is the Poisson distribution Equation 1 and the summation excludes the term $r = g = 0$. Note that if the distribution of tasks changes in time, then $\overline{\gamma}_{r,g}$ are time-dependent, $\overline{\gamma} = \overline{\gamma}_{r,g}(t)$.

To proceed further, we need to evaluate the summations in Equation 5. Let us define an auxiliary function

$$F(x) = \sum_{m_r=0}^{\infty}\sum_{m_g=0}^{\infty} x^{m_r+m_g}\frac{\lambda_r^{m_r}\lambda_g^{m_g}}{m_r!m_g!}e^{-\lambda_r}e^{-\lambda_g}\frac{m_r}{m_r+m_g} \tag{6}$$

It is easy to check that $\overline{\gamma}_{r,g}$ are given by

$$\begin{aligned}\overline{\gamma}_r &= F(1) - \frac{1}{2}P(0,0) = F(1) - \frac{1}{2}e^{\alpha h M_0} \\ \overline{\gamma}_g &= 1 - F(1) - \frac{1}{2}e^{\alpha h M_0}\end{aligned} \tag{7}$$

Differentiating Equation 6 with respect to x yields

$$\frac{dF}{dx} = \sum_{m_r=1}^{\infty}\sum_{m_g=0}^{\infty} x^{m_r+m_g-1}\frac{\lambda_r^{m_r}\lambda_g^{m_g}}{m_r!m_g!}e^{-\lambda_r}e^{-\lambda_g}m_r \tag{8}$$

Note that the summation over m_r starts from $m_r = 1$. Clearly, the sums over m_r and m_g are de–coupled thanks to the cancellation of the denominator (m_r+m_g):

$$\frac{dF}{dx} = \left(e^{-\lambda_r}\sum_{m_r=1}^{\infty} x^{m_r-1}\frac{\lambda_r^{m_r}}{m_r!}m_r\right)\left(e^{-\lambda_g}\sum_{m_g=0}^{\infty}\frac{(x\lambda_g)^{m_g}}{m_g!}\right) \tag{9}$$

The resulting sums are evaluated easily (as the Taylor expansion of corresponding exponential functions), and the results is

$$\frac{dF}{dx} = \lambda_r e^{-\lambda_0(1-x)} \tag{10}$$

where $\lambda_0 = \lambda_r + \lambda_g$. After elementary integration of Equation 10 (subject to the condition $F(0) = 1/2$), we obtain using Equation 8 and the expressions for λ_r, λ_0:

$$\overline{\gamma}_{r,g}(t) = \frac{1 - e^{\alpha h M_0}}{h}\int_{t-h}^{t} dt' \mu_{r,g}(t') \tag{11}$$

Here $\mu_{r,g}(t) = M_{r,g}(t)/M_0$ are the fraction of *Red* and *Green* tasks respectively.

Let us first consider the case when the task distribution does not change with time, *i.e.*, $\mu_r(t) = \mu_0$. Then we have

$$\overline{\gamma}_{r,g}(t) = (1 - e^{-\alpha h M_0})\mu^0_{r,g} \tag{12}$$

The solution of Equation 4 subject to the initial condition $p_r(t=0) = p_0$ is readily obtained:

$$p_r(t) = \mu^0_r + \left(p_0 - \frac{\overline{\gamma}_r}{\overline{\gamma}_r + \overline{\gamma}_g}\right) e^{-\varepsilon(\overline{\gamma}_r + \overline{\gamma}_g)t} \tag{13}$$

One can see that the probability distribution approaches the desired steady state value $p^s_r = \mu^0_r$ exponentially. Also, the coefficient of the exponent depends on the density of tasks and the length of the history window. Indeed, it is easy to check that $\overline{\gamma}_r + \overline{\gamma}_g = 1 - e^{-\alpha h M_0}$. Hence, for large enough M_0 and h, $\alpha h M_0 \gg 1$, the convergence rate is determined solely by ε. For a small task density or short history length, on the other hand, the convergence rate is proportional to the number of tasks, $\varepsilon(1 - e^{-\alpha h M_0}) \sim \varepsilon \alpha h M_0$. Note that this is a direct consequence of the rule that robots do not change their state whenever there are no observation in the history window.

Now let us consider the case where the task distribution changes suddenly at time t_0, $\mu_r(t) = \mu^0_r + \Delta\mu\theta(t - t_0)$, where $\theta(t - t_0)$ is the step function. For simplicity, let us assume that $\alpha h M_0 \gg 1$ so that the exponential term in Equation 11 can be neglected,

$$\overline{\gamma}_{r,g}(t) = \frac{1}{h}\int_{t-h}^{t} dt' \mu_{r,g}(t'), \overline{\gamma}_r(t) + \overline{\gamma}_g = 1 \tag{14}$$

Replacing Equation 14 into Equation 4, and solving the resulting differential equation yields

$$\begin{aligned}
p_r(t) &= \mu^0_r + \frac{\Delta\mu}{h}t - \frac{\Delta\mu}{\varepsilon h}(1 - e^{-\varepsilon t}), \qquad && t \le h \\
p_r(t) &= \mu^0_r + \Delta\mu - \frac{\Delta\mu}{\varepsilon h}(e^{-\varepsilon(t-h)} - e^{-\varepsilon t}), && t > h\,.
\end{aligned} \tag{15}$$

Equation 15 describes how the robot distribution converges to the new steady state value after the change in task distribution. Clearly, the convergence properties of the solutions depend on h and ε. It is easy to see that in the limiting case $\varepsilon h \gg 1$ the new steady state is attained after time h, $|p_r(h) - (\mu_0 + \Delta\mu)| \sim \Delta\mu/(\varepsilon h) \ll 1$, so the convergence time is $t_{conv} \sim h$. In the other limiting case $\varepsilon h \ll 1$, on the other hand, the situation is different. A simple analysis of Equation 15 for $t > h$ yields $|p_r(t) - (\mu_0 + \Delta\mu)| \sim \Delta\mu e^{-\varepsilon t}$ so the convergence is exponential with characteristic time $t_{conv} \sim 1/\varepsilon$.

4.1.3 Collective Behavior

In order to make predictions about the behavior of an MRS using a dynamic task allocation mechanism, we need to develop a mathematical model of the collective behavior of the system. In the previous section we derived a model of how an individual robot's behavior changes in time. In this section we extend it to model the behavior of a MRS. In particular, we study the collective behavior of a homogenous system consisting of N robots with identical controllers. Mathematically, the MRS is described by a probability density function that includes the states of all N robots. However, in most cases we are interested in studying the evolution of global, or average, quantities, such as the average number of robots in the *Red* state, rather than the exact probability density function. This applies when comparing theoretical predictions with results of experiments, which are usually quoted as an average over many experiments. Since the robots in either state are independent of each other, $p_r(t)$, is now the fraction of robots in the *Red* state, and consequently $Np_r(t)$ is the average number of robots in that state. The results of the previous section, namely solutions for $p_r(t)$ for constant task distribution (Equation 13) and for changing task distribution (Equation 15), can be used to study the average collective behavior. Section 6.1 presents results of analysis of the mathematical model.

4.1.4 Stochastic Effects

In some cases it is useful to know the probability distribution of robot task states over the entire MRS. This probability function describes the exact collective behavior from which one could derive the average behavior as well as the fluctuations around the average. Knowing the strength of fluctuations is necessary for assessing how the probabilistic nature of robot's observations and actions affects the global properties of the system. Below we consider the problem of finding the probability distribution of the collective state of the system.

Let $P_n(t)$ be the probability that there are exactly n robots in the *Red* state at time t. For a sufficiently short time interval Δt we can write [15]

$$P_n(t+\Delta t) = \sum_{n'} W_{n'n}(t;\Delta t)P_{n'}(t) - \sum_{n'} W_{nn'}(t;\Delta t)P_n(t) \tag{16}$$

where $W_{ij}(t;\Delta t)$ is the transition probability between the states i and j during the time interval $(t, t+\Delta t)$. In our MRS, this transitions correspond to robots changing their state from *Red* to *Green* or vice versa. Since the probability that more than one robot will have a transition during a time interval Δt is $O(\Delta t)$, then, in the limit $\Delta t \to 0$ only transitions between neighboring states are allowed in Equation 16, $n \to n \pm 1$. Hence, we obtain

$$\frac{dP_n}{dt} = r_{n+1}P_{n+1}(t) + g_{n-1}P_{n-1}(t) - (r_n + g_n)P_n(t)\,. \tag{17}$$

Here r_k is the probability density of having one of the k *Red* robots change its state to *Green*, and g_k is the probability density of having one of the $N-k$ *Green* robots change its state to *Red*. Let us assume again that $\alpha h M_0 \gg 1$ so that $\overline{\gamma}_g = 1 - \overline{\gamma}_r$. Then one has

$$r_k = k(1-\overline{\gamma}_r)\ ,\ g_k = (N-k)\overline{\gamma}_r \tag{18}$$

with $r_0 = g_{-1} = 0$, $r_{N+1} = g_N = 0$. $\overline{\gamma}_r$ is history-averaged transition rate to *Red* states.

The steady state solution of Equation 17 is given by [10]

$$P_n^s = \frac{g_{n-1}g_{n-2}...g_1g_0}{r_nr_{n-1}...r_2r_1}P_0^s \tag{19}$$

where P_0^s is determined by the normalization:

$$P_0^s = \left[1 + \sum_{n=1}^{N}\frac{g_{n-1}g_{n-2}...g_1g_0}{r_nr_{n-1}...r_2r_1}\right]^{-1} \tag{20}$$

Using the expression for $\overline{\gamma}$, after some algebra we obtain

$$P_n^s = \frac{N!}{(N-n)!n!}\overline{\gamma}_r^n(1-\overline{\gamma}_r)^{N-n} \tag{21}$$

e.g., the steady state is a binomial distribution with parameter $\overline{\gamma}$. Note again that this is a direct consequence of the independence of the robots' dynamics. Indeed, since the robots act independently, in the steady state each robot has the same probability of being in either state. Moreover, using this argument it becomes clear that the time-dependent probability distribution $P_n(t)$ is given by Equation 21 with $\overline{\gamma}$ replaced by $p_r(t)$, Equation 13.

4.2 Observations of Tasks and Robots

In this section we study the more complex dynamic task allocation mechanism in which robots make decisions to change their state based on the observations of not only available tasks but also on the observed task states of other robots. Specifically, each robot now records the numbers and types of task as well as the numbers and task types of robots it has encountered. Again, we let m_r and m_g be the number of tasks of *Red* and *Green* type, and n_r and n_g be the number of robots in *Red* and *Green* task state in a robot's history window. The probabilities for changing a robot's state are again given by transition functions that now depend on the fractions of observed tasks and robots of each type: $\hat{m}_r = m_r/(m_r + m_g)$, $\hat{m}_g = m_g/(m_r + m_g)$, $\hat{n}_r = n_r/(n_r + n_g)$, and $\hat{n}_g = n_g/(n_r + n_g)$. In our previous work [15] we showed that in order to achieve the desired long term behavior for task allocation (*i.e.*, in the steady state the average fraction of *Red* and *Green* robots is equal to the fraction of *Red* and *Green* tasks respectively), the transition rates must have the following functional form:

$$f_{g\to r}(\hat{m}_r, \hat{n}_r) = \hat{m}_r g(\hat{m}_r - \hat{n}_r), \tag{22}$$
$$f_{r\to g}(\hat{m}_r, \hat{n}_r) = \hat{m}_g g(\hat{m}_g - \hat{n}_g) \equiv (1-\hat{m}_r)g(-\hat{m}_r + \hat{n}_r). \tag{23}$$

Here $g(z)$ is a continuous, monotonically increasing function of its argument defined on an interval $[-1, 1]$. In this paper we consider the following forms for $g(z)$:

- *Power:* $g(z) = 100^z/100$
- *Stepwise linear:* $g(z) = z\Theta(z)$.[1]

To analyze this task allocation model, let us again consider a single robot that searches for tasks to perform and makes a transition to *Red* and *Green* states according to transition functions defined above. Let $p_r(t)$ be the probability that the robot is in the *Red* state at time t, with Equation 2 governing its time evolution. Note that $p_r(t)$ is also the average fraction of *Red* robots, $p_r(t) = N_r(t)/N$.

As in the previous case, the next step of the analysis is averaging over the the robot's histories, *i.e.*, $\hat{m}_r$ and $\hat{n}_r$. Note that a robot's observations of available tasks can still be modeled by a Poisson distribution similar to Equation 1. However, since the number of robots of each task state changes stochastically in time, the statistics of n_r and n_g should be modeled as a doubly stochastic Poisson process (also called Cox process) with stochastic rates. This would complicate the calculation of the average over $\hat{n}_r = n_r/(n_r + n_g)$ and require mathematical details that go well beyond the scope of this paper. Fortunately, as we demonstrated in the previous section, if a robot's observation window contains many readings, then the estimated fraction of task types is exponentially close to the average of the Poisson distribution. This suggests that for sufficiently high densities of tasks and robots we can neglect the stochastic effects of modeling observations for the purpose of our analysis, and replace the robot's observation by their average (expected) values. In other words, we use the following approximation:

$$\hat{n}_r \approx \frac{1}{h}\int_{t-h}^{t} p_r(t')dt' \tag{24}$$

$$\hat{m}_r \approx \frac{1}{h}\int_{t-h}^{t} \mu_r(t')dt'. \tag{25}$$

The Equations 2, 24, and 25 are a system of integro–differential equations that uniquely determine the dynamics of $p_r(t)$. In the most general case it is not possible to obtain solutions by analytical means, hence one has to solve the system numerically. However, if the task density does not change in time, we can still perform steady state analysis. Steady state analysis looks for long-term solutions that do not change in time, *i.e.*, $dp_r/dt = 0$. Let μ_r^0 be the density of *Red* tasks, and $p_0 = p_r(t \to \infty)$ be the steady state value, so that $\hat{m}_r = \mu_r^0$, $\hat{n}_r = p_r^0$. Then, by setting left hand side of Equation 2 to zero, we get

$$(1 - p_0)\mu_r^0 g(\mu_r^0 - p_0) = p_0(1 - \mu_r^0)g(-\mu_r^0 + p_0) \tag{26}$$

Note that $p_0 = \mu_r^0$ is a solution to Equation 26 so that in the steady state the fraction of *Red* robots equals the fraction of red tasks as desired. To show that this is the only solution, we note that for a fixed μ_r^0 the right- and left-hand

[1] The step function Θ is defined as $\Theta(z) = 1$ if $z \geq 0$; otherwise, it is 0. The step function guarantees that no transitions to *Red* state occur when $m_r < n_r$.

sides of the equation are monotonically increasing and decreasing functions of p_0 respectively, due to the monotonicity of $g(z)$. Consequently, the two curves can meet only once and that proves the uniqueness of the solution.

4.2.1 Phenomenological Model

Exact stochastic models of task allocation can quickly become analytically intractable, as we saw above. Instead of exact models, it is often more convenient to work with the so-called Rate Equations model. These equations can be derived from the exact stochastic model by appropriately averaging it [15]; however, they are often (see, for example, population dynamics [6]) phenomenological, or *ad hoc*, in nature — constructed by taking into account the system's salient processes. This approach makes a number of simplifying assumptions: namely, that the system is uniform and dilute (not too dense), that actions of individual entities are independent of one another, that parameters can be represented by their mean values and that system behavior can be described by its average value. Despite these simplifications, resulting models have been shown to correctly describe dynamics of collective behavior of robotic systems [18]. Phenomenological models are useful for answering many important questions about the performance of a MRS, such as, does the steady state exist, how long does it take to reach it, and so on. Below we present a phenomenological model of dynamic task allocation.

Individual robots are making their decisions to change task state probabilistically and independently of one another. A robot will change state from *Green* to *Red* with probability $f_{g\to r}$ and with probability $1 - f_{g\to r}$ it will remain in the *Green* state. We can succinctly write $\Delta N_{g\to r}$ and $\Delta N_{r\to g}$, the number of robots that switch from *Green* to *Red* and *vice versa* during a sufficiently small time interval Δt, as

$$\begin{aligned}
\Delta N_{g\to r} &= \sum_{i=1}^{N_g} x_i \big(f_{g\to r}\delta(x_i - 1) + (1 - f_{g\to r})\delta(x_i)\big) \\
\Delta N_{r\to g} &= \sum_{i=1}^{N_r} (1 - x_i)\big(f_{r\to g}\delta(x_i) + (1 - f_{r\to g})\delta(x_i - 1)\big) .
\end{aligned}$$

Here we introduced a state variable x_i, such that $x_i = 1$ when a robot is in the *Green* state, and $x_i = 0$ when a robot is in the *Red* state. $\delta(x)$ is Kronecker delta, defined as $\delta(x) = 1$ when $x = 0$ and $\delta(x) = 0$ otherwise. Therefore, $\Delta N_{g\to r}$ is a random variable from a binomial distribution specified by a mean $\mu = f_{g\to r}N_g$ and variance $\sigma^2 = f_{g\to r}(1 - f_{g\to r})N_g$. Similarly, the distribution of the random variable $\Delta N_{r\to g}$ is specified by mean $\mu = f_{r\to g}N_r$ and variance $\sigma^2 = f_{r\to g}(1 - f_{r\to g})N_r$.

During a time interval Δt the total number of robots in *Red* and *Green* task states will change as individual robots make decisions to change states. The following finite difference equation specifies how the number of *Red* will change on average:

$$N_r(t+\Delta t) = N_r(t) + \varepsilon\Delta N_{g\to r}\Delta t - \varepsilon\Delta N_{r\to g}\Delta t \tag{27}$$

Rearranging the equation and taking the continuous time limit ($\Delta t \to 0$) yields a differential Rate Equation that describes time evolution of the number of *Red* robots. By taking the means of ΔN's as their values, we recover Equation 2.

Keeping ΔN's as random variables allows us to study the effect the probabilistic nature of the robots' decisions have on the collective behavior.[2] We solve Equation 27 by iterating it in time and drawing ΔN's at random from their respective distributions. The solutions are subject to the initial condition $N_r(t \le 0) = N$ and specify the dynamics of task allocation in robots.

Functions $f_{g\to r}$ and $f_{r\to g}$ are calculated using estimates of the densities of *Red* tasks (m_r) and robots in *Red* state (n_r) from the observed counts stored in the robot's history window.

Transition rates $f_{g\to r}$ and $f_{r\to g}$ in the model are mean values, averaged over all histories and all robots. In order to compute them, we need to aggregate observations of all robots. Suppose each robot has a history window of length h. For a particular robot i, the values in the most recent observational slot are $N^0_{i,r}$, $N^0_{i,g}$, $M^0_{i,r}$ and $M^0_{i,g}$, the observed numbers of *Red* and *Green* robots and tasks respectively at time t. In the next latest slot, the values are $N^1_{i,r}$, $N^1_{i,g}$, $M^1_{i,r}$ and $M^1_{i,g}$, the observed numbers at time $t-\Delta$, and so on. Each robot estimates the densities of *Red* robots and tasks using the following calculation:

$$n_{i,r} = \frac{1}{h}\sum_{j=0}^{h-1}\frac{N^j_{i,r}}{N^j_{i,r}+N^j_{i,g}} = \frac{1}{h}\sum_{j=0}^{h-1} n^j_{i,r} \tag{28}$$

$$m_{i,r} = \frac{1}{h}\sum_{j=0}^{h-1}\frac{M^j_{i,r}}{M^j_{i,r}+M^j_{i,g}} = \frac{1}{h}\sum_{j=0}^{h-1} m^j_{i,r}. \tag{29}$$

When observations of all robots are taken into account, the mean of the observed densities of *Red* robots at time t — $\frac{1}{N}\sum_{i=1}^{N} n^0_{i,r}$ — will fluctuate due to observation noise, but on average it will be proportional to $N_r(t)/N$, which is the actual density of *Red* robots at time t. The proportionality factor is related to physical robot parameters, such as speed and observation area (see Section 6.1). Likewise, the average of the observed densities at time $t-j\Delta$ is $\frac{1}{N}\sum_{i=1}^{N} n^j_{i,r} \propto N_r(t-j\Delta)/N$, the density of robots at time $t-j\Delta$. Thus, the aggregate estimates of the fractions of *Red* robots and tasks are:

$$\hat{n}_r = \frac{1}{N}\sum_{i=1}^{N} n_{i,r} = \frac{1}{Nh}\sum_{j=0}^{h-1} N_r(t-j\Delta) \tag{30}$$

$$\hat{m}_r = \frac{1}{N}\sum_{i=1}^{N} m_{i,r} = \frac{1}{Mh}\sum_{j=0}^{h-1} M_r(t-j\Delta) \tag{31}$$

[2] Note that we do not model here the effect of observation noise due to uncertainty in sensor readings and fluctuations in the distribution of tasks.

Robots are making their decisions asynchronously, *i.e.*, at slightly different times. Therefore, the last terms in the above equations are best expressed in continuous form: *e.g.*, $1/Nh \int_h^0 N_r(t-\tau)d\tau$ (see Equation 24 and Equation 25).

Estimates Equation 30 and 31 can be plugged into Equation 22 and Equation 23 to compute the values of transition probabilities for any choice of the transition function (power or linear). Once we know $f_{r \to g}$ and $f_{g \to r}$, we can solve Equation 27 to study the dynamics of task allocation in robots. Note that Equation 27 is now a time-delay finite difference equation, and solutions will show typical oscillations.

We solve the models presented in this section and validate their predictions in context of the multi-foraging task described next.

5 Multi-Foraging Task in Robots

In this section we describe the multi-foraging task domain in which we experimentally tested our dynamic task allocation mechanism, including the simulation environment used and robot sensing and control characteristics. In Section 6.1 we use this application to validate the models presented above, solve them and compare their solutions to the results of embodied simulations.

5.1 Task Description

The traditional foraging task is defined by having an individual robot or group of robots collect a set of objects from an environment and either consume on the spot or return them to a common location [5]. Multi-foraging, a variation on traditional foraging, is defined in [2] and consists of an arena populated by multiple types of objects to be concurrently collected.

In our multi-foraging domain, there are two types of objects (e.g., pucks) randomly dispersed throughout the arena: Puck_{Red} and Puck_{Green} pucks that are distinguishable by their color. Each robot is equally capable of foraging both puck types, but can only be allocated to foraging for one type at any given time. Additionally, all robots are engaged in foraging at all times; a robot cannot be idle. A robot may switch the puck type for which it is foraging according to its control policy, when it determines it is appropriate to do so. This is an instantiation of the general task allocation problem described earlier in this paper, with puck colors representing different task types.

In the multi-foraging task, the robots move in an enclosed arena and pick up encountered pucks. When a robot picks up a puck, the puck is consumed (i.e., it is immediately removed from the environment, not transported to another region) and the robot carries on foraging for other pucks. Immediately after a puck is consumed, another puck of the same type is placed in the arena at a random location. This is done so as to maintain a constant puck density in the arena throughout the course of an experiment. In some situations, the density of pucks can impact the accuracy or speed of convergence to the desired task allocation. This is an important consideration in dynamic task allocation mechanisms for many domains; however, in this work we want to limit the number of experimental variables impacting system performance. Therefore, we reserve the investigation on the impact of varying puck densities for future work.

The role of dynamic task allocation in this domain requires the robots to split their numbers by having some forage for Puck_{Red} pucks and others for Puck_{Green} pucks. For the purpose of our experiments, we desire an allocation of robots to converge to a situation in which the proportion of robots foraging for Puck_{Red} pucks is equal to the proportion of Puck_{Red} pucks present in the foraging arena (e.g., if Puck_{Red} pucks make up 30% of the pucks present in the foraging arena, then 30% of the robots should be foraging for Puck_{Red} pucks). In general, the desired allocation could take other forms. For example, it could be related to the relative reward or cost of foraging each puck type without change to our approach.

We note that the limited sensing capabilities and lack of direct communication of the individual robots in the implementation of our task domain prohibits them from acquiring global information such as the size and shape of the foraging arena, the initial or current number of pucks to be foraged (total or by type), or the initial or current number of foraging robots (total or by foraging type).

5.2 Simulation Environment

In order to experimentally demonstrate the dynamic task allocation mechanism we made use of a physically-realistic simulation environment. Our simulation trials were performed using Player and Gazebo simulation environments. Player [3] is a server that connects robots, sensors, and control programs over a network. Gazebo [12] simulates a set of Player devices in a 3-D physically-realistic world with full dynamics. Together, the two represent a high-fidelity simulation tool for individual robots and teams that has been validated on a collection of real-robot robot experiments using Player control programs transferred directly to physical mobile robots. Figure 1 provides snapshots of the simulation environment used. All experiments involved 20 robots foraging in a 400m^2 arena.

The robots used in the experimental simulations are realistic models of the ActivMedia Pioneer 2DX mobile robot. Each robot, approximately 30 cm in diameter, is equipped with a differential drive, an odometry system using wheel rotation encoders, and 180 degree forward-facing laser rangefinder used for obstacle avoidance and as a fiducial detector/reader. Each puck is marked with a fiducial that marks the puck type and each robot is equipped with a fiducial that marks the active foraging state of the robot. Note that the fiducials do not contain unique identities of the pucks or robots but only mark the type of the puck or the puck type a given robot is engaged in foraging. Each robot is also equipped with a 2-DOF gripper on the front, capable of picking up a single 8 cm diameter puck at a time. There is no capability available for explicit, direct communication between robots nor can pucks and other robots be uniquely

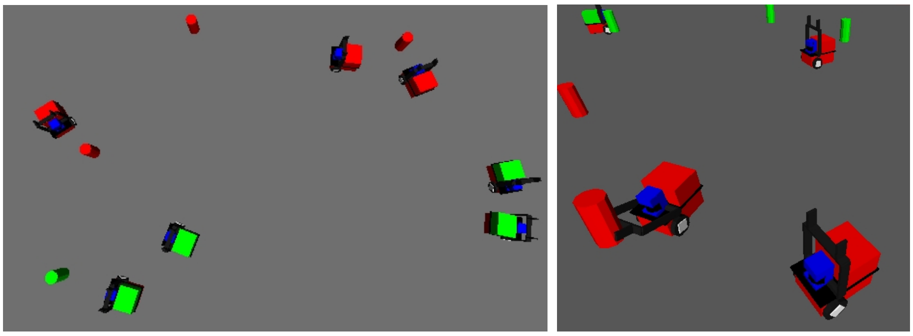


Figure 1: Snapshots from the simulation environment used. (left) An overhead view of foraging arena and robots. (right) A closeup of robots and pucks.

identified.

5.3 Behavior-Based Robot Controller

All robots have identical behavior-based controllers consisting of the following mutually exclusive behaviors: Avoiding, Wandering, Puck Servoing, Grasping, and Observing. Descriptions of robot behaviors are provided below.

- The **Avoiding** behavior causes the robot to turn to avoid obstacles in its path.
- The **Wandering** behavior causes the robot to move forward and, after a random length of elapsed time, to turn left or right through a random arc for a random period of time.
- The **Puck Servoing** behavior causes the robot to move toward a detected puck of the desired type. If the robot's current foraging state is Robot_{Red}, the desired puck type is Puck_{Red}, and if the robots current foraging state is Robot_{Green}, the desired puck type is Puck_{Green}.
- The **Grasping** behavior causes the robot to use its gripper to pick up and consume a puck within the gripper's grasp.
- The **Observing** behavior causes the robot to take the current fiducial information returned by the laser rangefinder and record the detected pucks and robots to their respective histories. The robot then updates its foraging state based on those histories. A description of the histories is given in Section 5.3.1 and a description of the foraging state update procedure is given in Section 5.3.2.

Each behavior listed above has a set of activation conditions based on relevant sensor inputs and state values. When met, the conditions cause the behavior to be become active. A description of when each activation condition is

Obstacle Detected	Puck_{Det} Detected	Gripper Break-Beam On	Observation Signal	Active Behavior
X	X	X	1	Observing
1	X	X	X	Avoiding
0	1	0	0	Puck Servoing
0	X	1	0	Grasping
0	X	X	X	Wandering

Table 1: Behavior Activation Conditions. Behaviors are listed in order of decreasing rank. Higher ranking behaviors preempt lower ranking behaviors in the event multiple are active. X denotes the activation condition is irrelevant for the behavior.

active is given below. The activation conditions of all behaviors are shown in Table 1.

- The **Obstacle Detected** activation condition is true when an obstacle is detected by the laser rangefinder within a distance of 1 meter. Other robots, pucks, and the arena walls are considered obstacles.
- The **Puck_{Det} Detected** activation condition is true if the robot's current foraging state is Robot_{Det} and a puck of type Puck_{Det} (where Det is *Red* or *Green*) is detected within a distance of 5 meters and within $\pm$ 30 degrees of the robot's direction of travel.
- The **Gripper Break-Beam On** activation condition is true if the break-beam sensor between the gripper jaws detects an object.
- The **Observation Signal** activation condition is true if the distance traveled by the robot according to odometry since the last time the **Observing** behavior was activated is greater than 2 meters.

5.3.1 Robot State Information

All robots maintain three types of state information: foraging state, observed puck history, and observed robot history. The foraging state identifies the type of puck the robot is currently involved in foraging. A robot with a foraging state of Robot_{Red} refers to a robot engaged in foraging Puck_{Red} pucks and a foraging state of Robot_{Green} refers to a robot engaged in foraging Puck_{Green} pucks. For simplicity, we will refer to both robot foraging states and puck types as *Red* and *Green*. The exact meaning will be clear in context.

Each robot is outfitted with a colored beacon passively observable by nearby robots which indicates the robot's current foraging state. The color of the beacon changes to reflect the current state – a red beacon for a foraging state of *Red* and a green beacon for foraging state *Green*. Thus, the colored beacon acts as a form of local, passive communication conveying the robot's current foraging state. All robots maintain a limited, constant-sized history storing the most recently observed puck types and another constant-sized history storing the foraging state of the most recently observed robots. Neither of these histories contains a unique identity or location of detected pucks or robots, nor does it store a time stamp of when any given observation was made. The history of observed pucks is limited to the last `MAX-PUCK-HISTORY` pucks observed and the history of the foraging states of observed robots is limited to the last `MAX-ROBOT-HISTORY` robots observed.

While moving about the arena, each robot keeps track of the approximate distance it has traveled by using odometry measurements. At every interval of 2 meters traveled, the robot makes an observation performed by the *Observing* behavior. This procedure is nearly instantaneous; therefore, the robot's behavior is not outwardly affected. The area in which pucks and other robots are visible is within 5 meters and ± 30 degrees in the robot's direction of travel. Observations are only made after traveling 2 meters because updating too frequently leads to over-convergence of the estimated puck and robot type proportions due to repeated observations of the same pucks and/or robots. On average, during our experiments, a robot detected 2 pucks and robots per observation.

5.3.2 Foraging State Transition Function

After a robot makes an observation, it re-evaluates and probabilistically changes its current foraging state given the newly updated puck and robot histories. The probability by which the robot changes its foraging state is defined by the transition function. We experimentally studied transition functions given by Equation 4, Equation 22 and Equation 23 with both *power* and *linear* forms. Below we present results of analysis and simulations and discuss the consequences the choice of the transition function has on system level behavior.

6 Analysis and Simulations Results

The mathematical models developed in Section 4 can be directly applied to the multi-foraging task if we map *Red* and *Green* tasks to *Red* and *Green* pucks and task states of robots to their foraging states. Model parameters, such as ε, α, etc, depend on physical realizations of the implementation and can be computed from details of the multi-foraging task as described below.

6.1 Observations of Pucks Only

First, we study the model of dynamic task allocation, presented in Section 4.1, where robots observe only pucks and make decision to switch foraging state according to the transition functions given by Equation 4. We compared theoretical predictions of the robots' collective behavior with results from simulations. We used Equation 13 and 15 to compute how the average number of robots in the *Red* state changes in time when the puck distribution is suddenly changed.

The parameter values were obtained from experiments. $p_0 = 1.0$ was the initial density of *Red* robots (of 20 total robots), $\mu_0 = 0.3$ was the initial *Red* puck density (of 50 total pucks), which remained constant until it was changed by the experimenter. The first change in puck density was $\Delta\mu = 0.5$, meaning that 80% of the pucks in the arena are now *Red*. The second change in puck density was $\Delta\mu = -0.3$, to 50% *Red* pucks.

ϵ is the rate at which robots make decisions to switch states. Robot traveled 2 m between observations at an average speed of 0.2 m/s; therefore, there are 10 s between observations, and $\varepsilon = 0.1$. h, the history length, is the number of pucks in the robot's memory. αM^0 is the rate at which robots encounter pucks. A robot makes an observation of its local environment at discrete time intervals. The area visible to the robot is $A_{vis} = (5\ m)^2\pi/6 = 13.09$, with 1/6 coming from the 60^o angle of view. The arena area is $A = 315\ m^2$; therefore, $\alpha M^0 = A_{vis}M^0/A = 2.1$. We studied the dynamics of the system for different history lengths h.

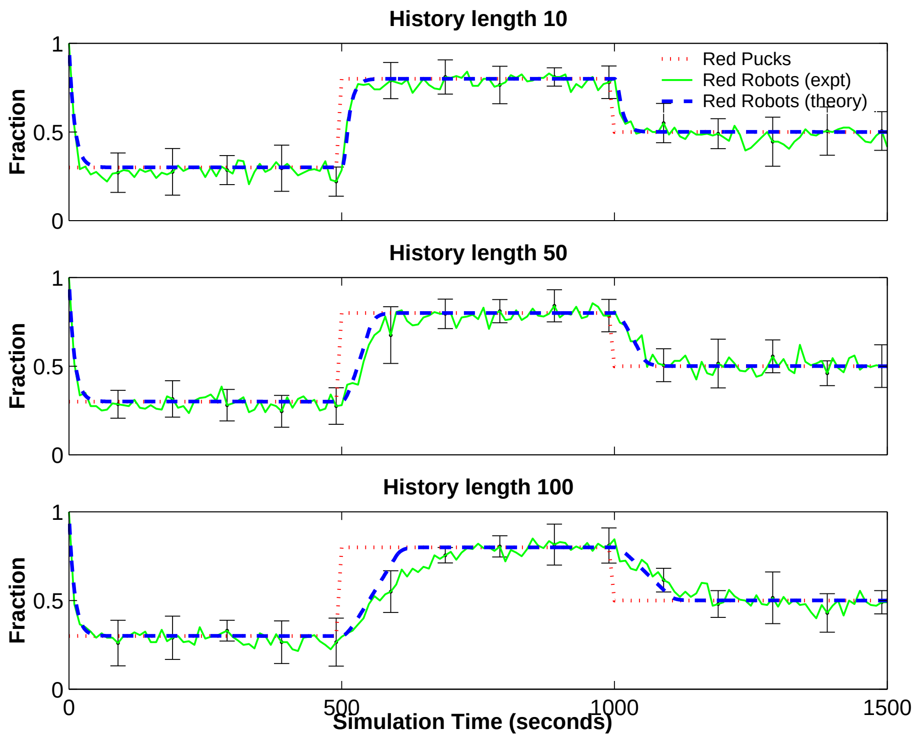


Figure 2: Evolution of the fraction of *Red* robots for different history lengths. Robots' decision to change state is based on observations of pucks only.

Figure 2 shows evolution of the numbers of *Red* robots for different history lengths. Initially, the distribution of *Red* pucks is set to 30% and all the robots are in the *Red* foraging state. At $t = 500\ s$, the puck distribution changes abruptly to 80%, and at $t = 1000\ s$ to 50%. The solid line shows results of simulations — the fraction of *Red* robots, averaged over 10 runs. The dashed line gives theoretical predictions for the parameters quoted above. Since we are in the $\varepsilon h \gg 1$ limit (for $h = 50, 100$), the time it takes to converge to the steady state is linear in history length, $t_{conv} \sim h$, as predicted by Equation 15. The agreement between theoretical and experimental results is excellent. We stress that there are no free parameters in the theoretical predictions — only experimental values of the parameters were used in producing these plots.

In addition to being able to predict the average collective behavior of the multi-robot system, we can also quantitatively characterize the amount of fluctuations in the system. Fluctuations are deviations from the steady state (after the system has converged to the steady state) that arise from the stochastic nature of robot's observations and decisions. These deviations result in fluctuations from the desired global distribution of *Red* and *Green* robots seen in an individual experiment. One can suppress these fluctuations by averaging results of many identical experiments.

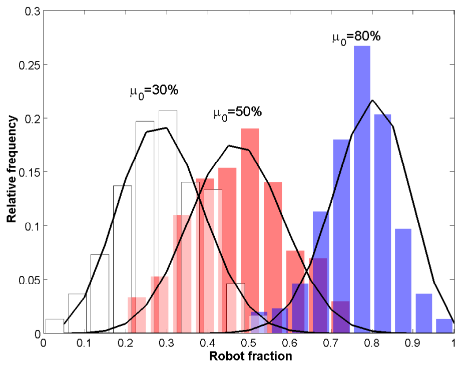


Figure 3: Histogram of the fraction of *Red* robots in the steady state for three different puck distributions (data for $h = 10$). μ_0 specifies fraction of *Red* pucks. Lines are theoretical predictions of the distribution of *Red* robots.

To measure the strength of the fluctuations, we take data from an individual experimental run and extract the fraction of *Red* robots, after the system has converged to the steady state, for each of the three *Red* puck distributions: $\mu_0 = 30\%,\ 50\%,\ 80\%$. Because the runs were relatively short, we only have $300\ s$ worth of data (30 data points) in the converged state; however, since each experiment was repeated ten times, we make the data sets longer by appending data from all experiments. In the end, we have 300 measurements of the steady state *Red* robot density for three different puck distributions. Figure 6.1 shows the histogram of robot distributions for three different puck distributions. The solid lines are computed using Equation 21, where for $\overline{\gamma}$ we used the actual means of the steady state distributions ($\overline{\gamma} = 0.28$, 0.47 and 0.7 for $\mu_0 = 30\%$, 50% and 80% respectively). We can see from the plots that the theory correctly predicts the strength of fluctuations about the steady state. As is true of binomial distributions, the fluctuations (measured by the variance) are greatest for cases where the numbers of *Red* and *Green* pucks are comparable ($\mu_0 = 50\%$) and smaller when their numbers are very different ($\mu_0 = 80\%$).

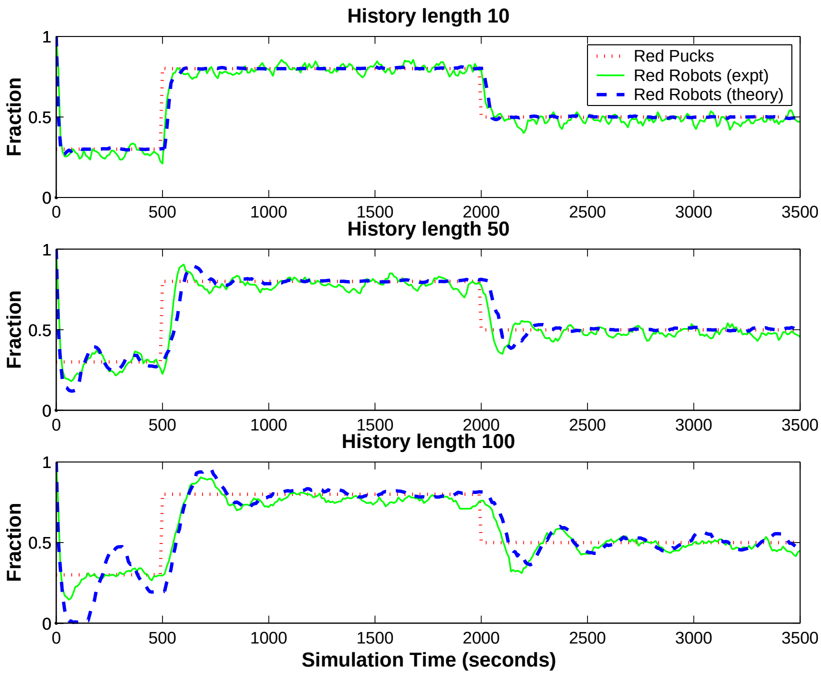
6.2 Observations of Pucks and Robots

In this section we study the dynamic task allocation model developed in Section 4.2, in which robots use observations of pucks and other robots' foraging states to make decision to change their own foraging state.

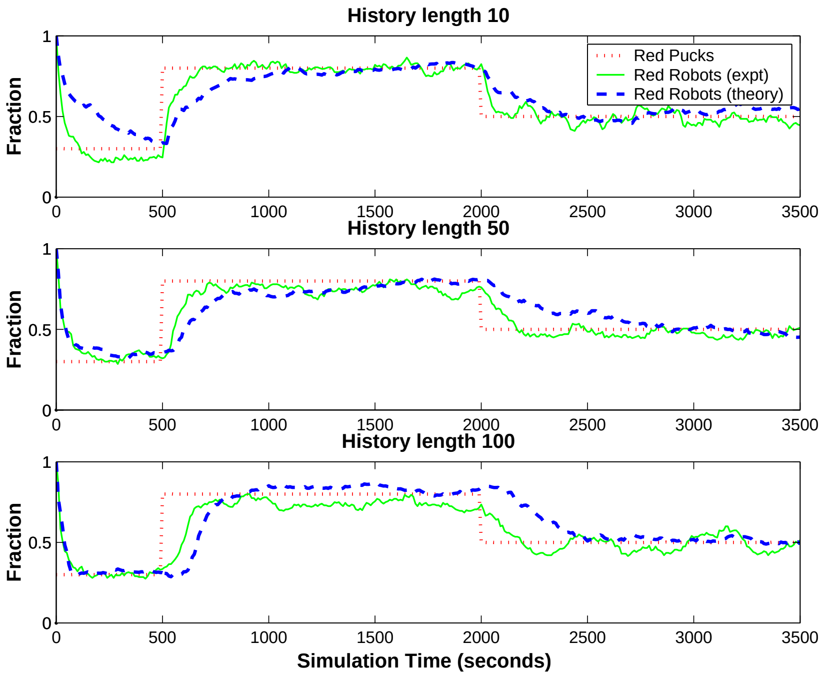
Figure 4 shows results of embodied simulations (solid lines) as well as solutions to the model Equation 27 (dashed lines) for different values of robot history length and forms of transition function (given by Equation 22 and 23, with $g(z)$ linear or power function). Initially, the *Red* puck fraction (dotted line) is 30%. It is changed abruptly at $t = 500\ s$ to 80% and then again at $t = 2000\ s$ to 50%. Each solid line showing *Red* robot density has been averaged over 10 runs. We rescale the dimensionless time of the model by parameter 10, corresponding $\varepsilon = 0.1$. The history length was the only adjustable parameter used in solving the equations. The values of h used to compute the observed fraction of *Red* robots n_r in Equation 30 were $h = 2,\ 8,\ 16$, corresponding to experimental history lengths $10,\ 50,\ 100$ respectively. For m_r, the observed fraction of *Red* pucks, we used their actual densities.

In order to explain the difference in history lengths between theory and experiment, we note that in the simulation experiments, the history length means the numbers of observed robots and pucks, while in the model, it means the number of observations, with multiple objects sighted within a single observation. According to calculations in Section 6.1, a robot observes about 2 pucks in a single observation. Moreover, the robot travels 2 m between observations, yet it sees 5 m out during each observation, meaning that individual observations will be correlated. Observations will be further correlated because of the pattern of a robot's motion — as the robot moves in a straight line towards a goal, it is likely to observe overlapping regions of the arena. These considerations could explain the factor of five difference between the history lengths used in the experiments and the corresponding values used in the model. More detailed experiments, for example, ones in which robots travel farther between observations, are necessary to explain these differences.

Solutions exhibit oscillations, although eventually oscillations decay and solutions relax to their steady state values. In all cases, the steady state value is the same as the fraction of red pucks in the arena. History-induced oscillations are far more pronounced for the linear transition function (Figure 4(a)) than for the power transition function (Figure 4(b)). For the power transition function, these oscillations are present but become evident only for longer history lengths. This behavior is probably caused by the differences between the values of transition functions near the steady state: while the value of the power transition function remains small near the steady state, the value of the linear transition function grows linearly with the distance from the steady state, thereby ampli-



(a) Linear transition function



(b) Power transition function

Figure 4: Evolution of the fraction of *Red* robots for different history lengths and transition functions, compared to predictions of the model

fying any deviations from the steady state solution. The amplitude and period of oscillations and the convergence rate of solutions to the steady state all depend on history length, and it generally takes longer to reach the steady state for longer histories. Another conclusion is that the linear transition function converges to the desired distribution faster than the power function, at least for moderate history lengths.

7 Discussion

We have constructed and analyzed mathematical models of dynamic task allocation in a multi-robot system. The models are general and 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. These models are based on theory of stochastic processes. In order to study a robot's behavior, we do not need to know its exact trajectory or the trajectories of other robots; instead, we derive a probabilistic model that governs how a robot's behavior changes in time. In some simple cases these models can be solved analytically. However, stochastic models are usually too complex for exact analytic treatment. Thus, in the scenario described in Section 4.1 in which only observations of tasks are made, though the individual model is tractable, the stochastic model of the collective behavior is not. Instead, we use averaging and approximation techniques to quantitatively study the dynamics of the collective behavior. Such models, therefore, do not describe the robots' behavior in a single experiment, but rather the behavior that has been averaged over many experimental or simulations runs. Fortunately, results of experiments and simulations are usually presented as an average over many runs; therefore, mathematical models of average collective behavior can be used to describe experimental results. In fact, the stochastic model produces excellent agreement with experimental results under all experimental conditions and without using any adjustable parameters.

Phenomenological models are more straightforward to construct and analyze than exact stochastic models — in fact, they can be easily constructed from details of the individual robot controller [18]. The ease of use comes at a price, namely, the number of simplifying assumptions that were made in order to produce a mathematically tractable model. First, we assume that the robots are functioning in a dilute limit, where they are sufficiently separated that their actions are largely independent of one another. Second, we assume that the transition rates can be represented by aggregate quantities that are spatially uniform and independent of the details of the individual robot's actions or history. We also assume the system is homogeneous, with modeled robots characterized by a set of parameters, each of them representing the mean value of some real robot feature: mean speed, mean duration for performing a certain maneuver, and so on. Real robot systems are heterogeneous: even if the robots are executing the same controller, there will always be variations due to inherent differences in hardware. We do not consider parameter distributions in our models as would be necessary to describe such heterogeneous systems. Finally, phenomenological models more reliably describe systems where fluctuations (deviations from the mean behavior) can be neglected, as happens in large systems or when many experimental runs are aggregated. However, even if phenomenological models don't agree with experiments exactly, as we saw in Section 6.2, they can still reliably predict most behaviors of interest even in not-so-large systems. They are, therefore, a useful tool for modeling and analyzing multi-robot systems.

8 Conclusion

Mathematical analysis can be a useful tool for the study and design of MRS and a viable alternative to experiments and simulations. It can be applied to large systems that are too costly to build or take too long to run in simulation. Mathematical analysis can be used to study the behavior of an MRS, select parameters that optimize its performance, prevent instabilities, *etc.* In conjunction with the design process, mathematical analysis can help understand the effect individual robot characteristics have on the collective behavior *before* a system is implemented in hardware or in simulation. Unlike experiments and simulations, where exhaustive search of the design parameter space is often required to reach any conclusion, analysis can often produce exact analytic results, or scaling relationships, for the quantities of interest. If these are not possible, exhaustive search of the parameter space is much more practical and efficient. Finally, results of analysis can be used as feedback to guide performance-enhancing modifications of the robot controller.

In this paper we have described an dynamic task allocation mechanism where robots use local observations of the environment to decide their task assignments. We have presented a mathematical model of this task allocation mechanism and studied it in the context of a multi-foraging task scenario. We compared predictions of the model with results of embodied simulations and found excellent quantitative agreement. In this application, mathematical analysis could help the designer choose robot properties, such as the form of the transition probability used by robots to switch their task state, or decide how many observations the robot ought to consider.

Mathematical analysis of MRS is a new field, but its success in explaining experimental results shows it to be a promising tool for the design and analysis of robotic systems. The field is open to new research directions, from applying analysis to new robotic systems to developing increasingly sophisticated mathematical models that, for example, account for heterogeneities in robot population that are due to differences in their sensors and actuators.

Acknowledgment

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-00-2-0573.

References

[1] William Agassounon and Alcherio 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, Hammamet, Tunisia.*, pages 250–255. oct 2002.

[2] Tucker Balch. The impact of diversity on performance in multi-robot foraging. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 92–99, Seattle, WA, 1999. ACM Press.

[3] B. Gerkey, R. Vaughan, K. Stoey, A. Howard, G. Sukhatme, and M. Matarić. Most valuable player: A robot device server for distributed control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1226–1231, Maui, Hawaii, Oct 2001.

[4] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International journal of robotics research*, 23(9):939–954, 2004.

[5] Dani Goldberg and Maja J Matarić. Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. In Tucker Balch and Lynne E. Parker, editors, *Robot Teams: From Diversity to Polymorphism*, pages 315–344. AK Peters, 2002.

[6] Richard Haberman. *Mathematical Models: Mechanical Vibrations, Population Dynamics, and Traffic Flow.* Society of Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[7] Bernardo A. Huberman and Tad Hogg. The behavior of computational ecologies. In B. A. Huberman, editor, *The Ecology of Computation*, pages 77–115, Amsterdam, 1988. Elsevier (North-Holland).

[8] A. J. Ijspeert, A. Martinoli, A. Billard, and L. M. Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171, 2001.

[9] Chris V. Jones and Maja J Matarić. Adaptive task allocation in large-scale multi-robot systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'03), Las Vegas, NV*, pages 1969–1974, oct 2003.

[10] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier Science, Amsterdam, revised and enlarged edition edition, 1992.

[11] Sanza Kazadi, A. Abdul-Khaliq, and Ron Goodman. On the convergence of puck clustering systems. *Robotics and Autonomous Systems*, 38(2):93–117, 2002.

[12] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.

[13] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.

[14] Kristina Lerman and Aram Galstyan. Agent Memory and Adaptation in Multi-Agent Systems. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2003), Melbourne, Australia*, pages 797–803, jul 2003.

[15] Kristina Lerman and Aram Galstyan. Macroscopic Analysis of Adaptive Task Allocation in Robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS-2003), Las Vegas, NV*, pages 1951–1956, oct 2003.

[16] Kristina Lerman and Aram Galstyan. Collectives and design of complex systems. chapter Two Paradigms for the Design of Artificial Collectives, pages 231–256. Springer Verlag, New York, 2004.

[17] Kristina Lerman, Aram Galstyan, Alcherio Martinoli, and Auke Ijspeert. A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life Journal*, 7(4):375–393, 2001.

[18] Kristina Lerman, Alcherio Martinoli, and Aram Galstyan. A review of probabilistic macroscopic models for swarm robotic systems. In Sahin E. and Spears W., editors, *Swarm Robotics Workshop: State-of-the-art Survey*, number 3342 in LNCS, pages 143–152. Springer-Verlag, Berlin Heidelberg, 2005.

[19] A. Martinoli. *Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies*. PhD thesis, PhD Thesis No 2069, EPFL, 1999.

[20] A. Martinoli, A. J. Ijspeert, and L. M. Gambardella. A probabilistic model for understanding and comparing collective aggregation mechanisms. In Dario Floreano, Jean-Daniel Nicoud, and Francesco Mondada, editors, *Proceedings of the 5th European Conference on Advances in Artificial Life (ECAL-99)*, volume 1674 of *LNAI*, pages 575–584, Berlin, September 13–17 1999. Springer.

[21] Lynne Parker. Alliance: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[22] Y. Sato and J. P Crutchfield. Coupled replicator equations for the dynamics of learning in multiagent systems. *Physical Review*, E67:015206, 2003.

[23] Ken Sugawara and Masaki Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica*, D100:343–354, 1997.

[24] Ken Sugawara, Masaki Sano, Ikuo Yoshihara, and K. Abe. Cooperative behavior of interacting robots. *Artificial Life and Robotics*, 2:62–67, 1998.

[25] David Wolpert and Kagan Tumer. An introduction to collective intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center, 1999.