Analysis of a Stochastic Model of Adaptive Task Allocation in Robots

Aram Galstyan and Kristina Lerman

Information Sciences Institute
University of Southern California
Marina del Rey, California
galstyan@isi.edu, lerman@isi.edu

Abstract. Adaptation is an essential requirement for self–organizing multi–agent systems functioning in unknown dynamic environments. Adaptation allows agents, e.g., robots, to change their actions in response to environmental changes or actions of other agents in order to improve overall system performance, and remain robust even while a sizeable fraction of agents fails. In this paper we present and study a simple model of adaptation for the task allocation problem in a multi–robot system. In our model robots have to choose between two types of tasks, and the goal is to achieve desired task division without any explicit communication between robots. Robots estimate the state of the environment from repeated local observations and decide what task to choose based on these observations. We model robots and observations as stochastic processes and study the dynamics of individual robots and the collective behavior. We validate our analysis with numerical simulations.

1 Introduction

Adaptation is an essential requirement for multi–agent systems functioning in dynamic environments that cannot be fully known or characterized in advance. Adaptation allows agents, whether they are robots, modules in an embedded system or software components, to change their behavior in response to environmental changes and actions of other agents in order to improve overall system performance. Additionally, adaptation allows swarms, artificial systems composed of large numbers of agents, to remain robust in face of failure even by a sizeable fraction of agents. If each agent had instantaneous global knowledge of the environment, it could dynamically adapt to any changes in the environment, including actions of other agents. In most situations, however, such global knowledge is impractical or infeasible to obtain; therefore, one needs to devise different adaptation mechanisms based on partial, possibly noisy information about the state of the environment and, possibly, of other agents. Also, one would prefer a mechanism that would require little or no communication and/or negotiations between the agents.

Analysis is an important part of designing adaptive, self–organizing systems since it allows to understand global system properties given the behavior of individual entities and the rules of interactions between them. There are generally two options for the analysis of swarm behavior: experiment and simulation. Experiments with real agents, e.g., robots, allow the researcher to observe swarms under real conditions; however, experiments are very costly and time consuming. Simulations, such as sensor based simulations for robots, attempt to realistically model the environment, the robots' imperfect sensing of and interactions with it. Though simulations are much faster and less costly than experiments, they suffer from many of the same limitations, namely, they are still time consuming to implement, and systematically exploring the parameter space is often tedious. Mathematical analysis is an alternative to the time consuming and often costly experiments and simulations. Using mathematical analysis we can study dynamics of multi–robot systems, predict long term behavior of even very large systems, gain insight into system design, for instance what parameters determine group behavior and how individual robot characteristics affect the swarm. Additionally, mathematical analysis can be used to choose parameters that optimize the swarm's performance, prevent instabilities and so on. Note, however, that the mathematical analysis usually requires strong simplifications and should be validated by comparing its results with the results of the more realistic simulations (such as sensor based) and/or actual experiments with robots.

In this paper we present and analyze a simple stochastic model for adaptive task allocation in a team of robots, where robots have to forage for two distinct types of pucks, *Red* and *Green* scattered around the arena [5]. Each robot is able to collect pucks of a specific type, say *Red*: when a robot's foraging state is set to *Red*, it is searching for and collecting *Red* pucks. The goal of adaptive task allocation mechanism is to achieve a distribution of robots between two states that, over time, correctly reflects the pucks' prevalence. The robots have no global information about the number of pucks of either color, or the states other robots. Instead, the robots make repeated local observations of the environment, store them in the memory, and use them to decide between two states. We analyze our model thoroughly using stochastic Master equation that describes the evolution of macroscopic dynamics, and compare it to the the results of discrete time simulations. We demonstrate that our analytical approach fully reproduces the results of the numerical simulations, suggesting that it might be used as an efficient tool for analyzing the global behavior adaptive multi-agent systems.

2 Related Work

Mathematical analysis of the behavior of robots is a relatively new field with approaches and methodologies borrowed from other fields, such as mathematics, physics and biology. In recent years, a number of studies appeared that attempted to mathematically model and analyze collective behavior of distributed robot systems. These include analysis of the effect of collaboration in foraging [16, 17] and stick-pulling [9, 12] experiments, the effect of interference in robot foraging [7], and robot aggregation task [1, 6]. So far this type of analysis has been limited to simple reactive or behavior-based robots in which perception and action are tightly coupled. Such robots take input from sensors or behaviors and send output to actuators or other behaviors.[1] They make no use of memory or historic state information.

The role of learning in improving the performance of a multi-robot system has been addressed by several researchers. The RoboCup robot soccer domain provided a fruitful framework for introducing learning in the context of multi-agent and multi-robot systems. Several authors examined the use of reinforcement learning to learn basic soccer skills, coordination techniques [14] and game strategies [15]. Matarić [13] reviews research on learning in behavior-based robot systems, including learning behavior policies, models of the environment and behavior history. Goldberg and Matarić [2] present a framework for learning models of interaction dynamics in multi-robot systems. These models are learned on-line and used by robots to detect anomalies in system performance as well as to recover from these anomalies. Their work shares common foundation with ours: Markov processes as a model of interactions between robots . However, adaptation occurs as a result of the changing representation — the model of the interactions created and updated by robots — not as a result of changes in robot behaviors. Li *et al.* [10] introduced learning into collaborative stick pulling robots and showed in simulation that learning does improve system performance by allowing robots to specialize. No analysis of the collective behavior or performance of the system have been attempted in any of these studies.

Huberman and Hogg [3] 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. They constructed a mathematical model of the dynamics of such systems and studied them under variety of conditions, including imperfect knowledge and delayed information. Although the mechanism for adaptation is different, their approach, which they termed "computational ecology" is similar in spirit to ours, as it is based on the foundations of stochastic processes and models of average behavior.

3 Dynamic Task Allocation in Robots

Chris Jones and Maja Matarić presented an embodied simulation study of adaptive task allocation in a distributed robot system [5]. In their study, two distinct types of pucks, *Red* and *Green*, were scattered around the arena. Each robot could be tasked to collect pucks of a specific type, say *Red*. When a robot's foraging state is set to *Red*, it is searching for and collecting *Red* pucks. The robot can also recognize the foraging state of robots it sees. The robots have no *a priori* information about the shape of the arena, the number of pucks left in it or the number of foraging robots. The goal of adaptive task allocation is to design a robot controller that will allow robots to dynamically adjust their foraging type, so that the number of robots searching for *Red* and *Green* pucks will, over time, correctly reflect the pucks' prevalence.

[1] Robots that use timers to trigger actions can also be studied using this approach.

The memory-based mechanism for adaptive behavior suggested by Jones and Matarić is consistent with the biologically-inspired control paradigm that has become popular in the field of distributed robotics. In such systems, the goal is to design local interactions among robots or between robots and the environment that will lead to the desired collective behavior. The mechanism works as follows: As it wanders around the arena, robot counts the number of pucks of each type in the environment as well as the number of robots in each foraging state visible to it and adds these counts to memory. Since memory has a finite size new observations replace the oldest ones. Periodically, the robot uses values in memory to estimate the density of pucks and robots of each type, and changes its foraging state according to a certain transition function. The general idea is that a robot should switch its state to *Red* if there are fewer than necessary robots in the *Red* state and *vice versa* for *Green*.

In this paper we propose and study a slightly simplified model for task allocation, where the robots determine whether to make a transition to a new state based on the number of pucks of either types they encountered. Specifically, let m_r and m_g be the number of red and green pucks respectively that a robot has encountered in some time interval, so that the estimated fraction of red pucks is $\mu_r = m_r/(m_r + m_g)$. Then, the robot will choose the red and green states with probability μ_r and $1 - \mu_r$ respectively. Clearly, if the robots have global knowledge about the number of red and green pucks then this simple algorithm will achieve the desired distribution of the robots between the states. Hence, it is interesting to see how the incomplete knowledge about the environment affects this distribution, and in the case of dynamic environment (e.g., changing ratio of red and green pucks) what is its effect on the adaptive properties of the system.

4 Modelling Robots Observations

As we explained above, the transition rates between the states depend on robots' observations, or history (memory). In our model, this history comprises of the number of red and green pucks a robot has encountered during a time interval τ. Let us assume that the process of encountering a puck is a Poisson process 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 pucks in the arena. This simplification is based on the idea that robot's interactions with other robots and the environment is independent of the robot's actual trajectory, but are governed by probabilities determined by simple geometric considerations. This simplification has been shown to produce remarkably good agreements with experiments [11, 4].

Let M_r and M_g be the number of red and green pucks respectively, that generally can be time dependent, $M_r(t) + M_g(t) = M_0$. The probability that in the time interval $[t - \tau, t]$ the robot has encountered exactly m_r and m_g pucks is the product of two Poisson distributions:

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

where $\lambda_i = \alpha \int_{t-\tau}^{t} dt' M_i(t')$, $i = r, g$, are the means the of respective distributions. In the case when the puck distribution does not change in time one has $\lambda_i = \alpha M_i \tau$, $i = r, g$.

5 Individual Dynamics

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

Let us consider a single robot that forages for *Red* and *Green* pucks in a closed area and makes a transition to *Red* and *Green* states according to its observations. As a designer, we would like to define transition rules so that the fraction of time the robot spends in the *Red* (*Green*) foraging state be equal to the fraction of red (green) pucks. Let $p_r(t)$ be the probability that the robot is in the *Red* state. The equation governing its evolution reads

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

where ε is the rate at which the robot has to make a decision whether to switch it state, and $f_{g\to r}$ and $f_{r\to g}$ are the corresponding transitions probabilities between the states. As we explained above, these probabilities depend on the robot's history, e.g., the number of either types of pucks it has encountered during the time interval τ preceding the transition. Specifically, let m_r and m_g be the number of red and green pucks respectively that a robot has encountered in that time interval. Then we define transition rates as follows:

$$f_{g\to r} = \frac{m_r}{m_r + m_g} \equiv \gamma(m_r, m_g), f_{r\to g} = 1 - \gamma(m_r, m_g) \tag{3}$$

Eq.2 is a stochastic differential equation since the coefficients (transition rates) depend on random variables m_r and m_g. Moreover, since the robot's history changes gradually, then the values of the coefficients at different times are correlated, hence making the exact treatment very difficult. Here we propose to study the it within the *annealed* approximation. Namely, we neglect the 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 Eq. 1. Then, we can average Eq.2 over the histories to obtain

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

where $\overline{\gamma}$ is the history–averaged transition rate

$$\overline{\gamma} = \sum_{m_r=0}^{\infty} \sum_{m_g=0}^{\infty} P(m_r, m_g) \frac{m_r}{m_r + m_g} \tag{5}$$

and $P(m_r, m_g)$ is the Poisson distribution Eq. 1. Note that if the pucks distribution changes in time then $\overline{\gamma}$ is time–dependent, $\overline{\gamma} = \overline{\gamma}(t)$. The solution of Eq. 4 subject to the initial condition $p_r(t = 0) = p_0$ is readily obtained:

$$p_r(t) = p_0 e^{-\varepsilon t} + \varepsilon \int_0^t dt' \bar{\gamma}(t - t') e^{-\varepsilon t'} \tag{6}$$

To calculate $\overline{\gamma}(t)$ we define an auxiliary function

$$F(x) = \sum_{m_r=0}^{\infty} \sum_{m_g=0}^{\infty} x^{m_r+m_g} \frac{\lambda_r^{m_r} \lambda_g^{m_g}}{m_r! m_g!} e^{-\lambda_r} e^{-\lambda_g} \frac{m_r}{m_r + m_g} \tag{7}$$

so that $\overline{\gamma} = F(x = 1)$. Differentiating Eq. 7 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 decoupled 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 Eq. 10 (subject to the condition $F(0) = 1/2$), and using the expressions for λ_r, λ_0 we obtain

$$\overline{\gamma}(t) = \frac{1}{\tau} \int_{t-\tau}^{t} dt' \mu_r(t') + e^{-\alpha\tau M_0} \left(\frac{1}{2} - \frac{1}{\tau} \int_{t-\tau}^{t} dt' \mu_r(t') \right) \tag{11}$$

where $\mu_r(t) = M_r(t)/M_0$ is the fraction of red pucks. Eq. 6 and 11 fully determine the evolution of the dynamics of a single robot. To analyze its properties, let us first consider the case when the puck distribution does not change with time, $\mu_r(t) = \mu_0$. Then the we have

$$p_r(t) = \overline{\gamma} + (p_0 - \overline{\gamma}) e^{-\varepsilon t} \tag{12}$$

$$\overline{\gamma} = \mu_0 + e^{-\alpha\tau M_0} (1/2 - \mu_0) \tag{13}$$

Hence, the probability distribution approaches its steady state value $p_r^s = \overline{\gamma}$ exponentially. Note that for large enough $\alpha\tau M_0$ the second term in the expression for $\overline{\gamma}$ can be neglected so that the steady state attains the desired value $p_r^s \approx \mu_0$. For small values of $\alpha\tau M_0$ (i.e., small density of pucks or short history window), however, the desired steady state is not reached, and in the limit of very small $\alpha\tau M_0$ it attains the value 1/2 regardless of the actual puck distribution (we elaborate on this more in Section 7).

Now let us consider the case when there is a sudden change in the puck distribution at a certain time t_0, $\mu_r(t) = \mu_0 + \Delta\mu\theta(t - t_0)$, where $\theta(t)$ is the step function (without loss of generality we set $t_0 = 0$). Clearly, after some transient time, the distribution will converge to its new equilibrium value $\mu_0+\Delta\mu$ (we assume that $\alpha\tau M_0$ is sufficiently large so we can neglect the exponential correction to the steady state value). After some simple algebra, we obtain from Eq. 6 and 11

$$p_r(t) = \mu_0 + \frac{\Delta\mu}{\tau}t - \frac{\Delta\mu}{\varepsilon\tau}(1 - e^{-\varepsilon t}), \qquad t \le \tau$$
$$p_r(t) = \mu_0 + \Delta\mu - \frac{\Delta\mu}{\varepsilon\tau}(e^{-\varepsilon(t-\tau)} - e^{-\varepsilon t}), \quad t > \tau \tag{14}$$

Eqs. 14 describes how robot distribution converges to the new steady state value after the change in puck distribution. Clearly, the convergence properties of the solutions depend on τ and ε. It is easy to see that in the limiting case $\varepsilon\tau \gg 1$ the new steady state is nearly attained after time τ, $|p_r(\tau) - (\mu_0 + \Delta\mu)| \sim \Delta\mu/(\varepsilon\tau) \ll 1$, so the convergence time is $t_{conv} \sim \tau$. In the other limiting case $\varepsilon\tau \ll 1$, on the other hand, the situation is different. Indeed, a simple analysis of Eqs. 14 for $t > \tau$ yields $|p_r(t) - (\mu_0 + \Delta\mu)| \sim \Delta\mu e^{-\varepsilon t}$ so the convergence is exponential with characteristic time $t_{conv} \sim 1/\varepsilon$.

6 Collective Behavior

In this section we study the collective behavior of a homogenous system consisting of N robots with identical controllers described in the previous section. Specifically, we are interested in the global system properties, namely, average number of robots in the given states and the fluctuations around this average. Note that the average number of robots in the *Red* state is directly related to Eq. 4. Indeed, since the robots are in either state independent of each other, then $p_r(t)$ is simply the fraction of robots in the *Red* state, and consequently $Np_r(t)$ is the average number of robots in that state. Below we consider a more general problem of finding the probability distribution of having n robots in the red state.

Let $P_n(t)$ be the probability density that there are exactly n *Red* robots at time t. For a sufficiently short time interval Δt we can write [8]

$$P_n(t + \Delta t) = \sum_{n'} W_{n'n}(t; \Delta t)P_{n'}(t) - \sum_{n'} W_{nn'}(t; \Delta t)P_n(t) \tag{15}$$

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 multi robot systems, this transitions correspond to robots changing their state from red to green or vice versa. Since probability of having more than one robot to have a transition during a time interval Δt is $o(\Delta t)$, then, in the limit $\Delta t \to 0$ only transition between neighboring states are allowed in Eq. 15, $n \to n \pm 1$. Hence, we obtain

$$\frac{dP_n}{dt} = r_{n+1}P_{n+1}(t) + g_{n-1}P_{n-1}(t) - (r_n + g_n)P_n(t). \tag{16}$$

Here r_k is the probability density for having one of the k *Red* robots to changes its state to *Green*, and g_k is the probability density for having one of the $N-k$ *Green* robots to change their state to *Red*:

$$r_k = k(1-\overline{\gamma})\ ,\ g_k = (N-k)\overline{\gamma} \tag{17}$$

with $r_0 = g_{-1} = 0$, $r_{N+1} = g_N = 0$. Again, we have averaged the transition probabilities over the histories.

The steady state solution of Eq. 16 is given by [18]

$$P_n^s = \frac{g_{n-1}g_{n-2}...g_1g_0}{r_nr_{n-1}...r_2r_1}P_0^s \tag{18}$$

where P_0^s is determined by the normalization:

$$P_0^s = \left[1 + \sum_{n=1}^{N} \frac{g_{n-1}g_{n-2}...g_1g_0}{r_nr_{n-1}...r_2r_1}\right]^{-1} \tag{19}$$

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

$$P_n^s = \frac{N!}{(N-n)!n!}\overline{\gamma}^n(1-\overline{\gamma})^{N-n} \tag{20}$$

e.g., the steady state is a binomial distribution with parameter $\overline{\gamma}$. Note again that this is the direct consequence of the independence of robots' dynamics. Indeed, since the robots act independently, then in the steady state each of them 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 Eq. 20 with $\overline{\gamma}$ replaced by $p_r(t)$, Eq. 6.

7 Simulations

To validate our analytic mathematical model, we compared its predictions to the results of discrete time numeric simulations with 100 robots. We model the arena by a 100×100 rectangular grid, M_r (M_g) cells are occupied by red (green) pucks. Robots move randomly from cell to cell[2], and once they are on a cell with either type of puck, they record it in their register, or memory. At each time step, each robot, with probability ε decides whether it should consider a transition or not,

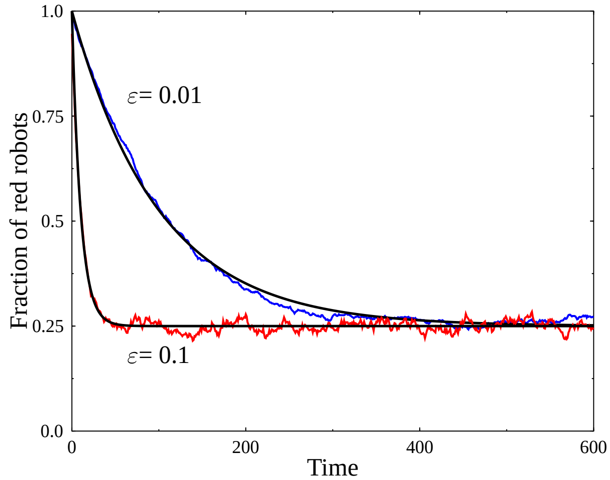

Fig. 1. Fraction of red robots vs time, $\tau = 50$

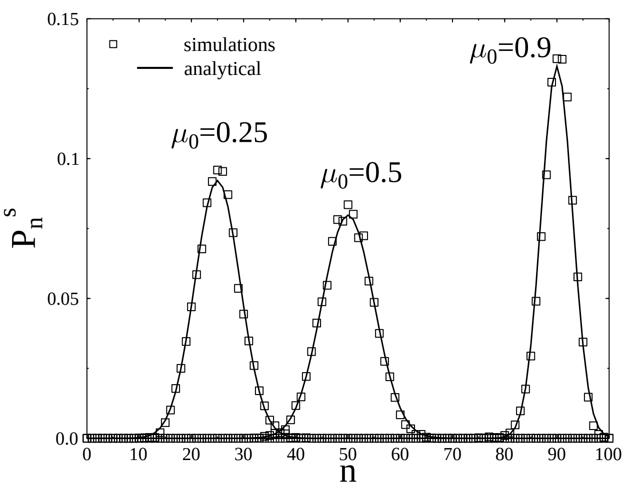

Fig. 2. Steady state distribution P_n^s for different fractions of red pucks

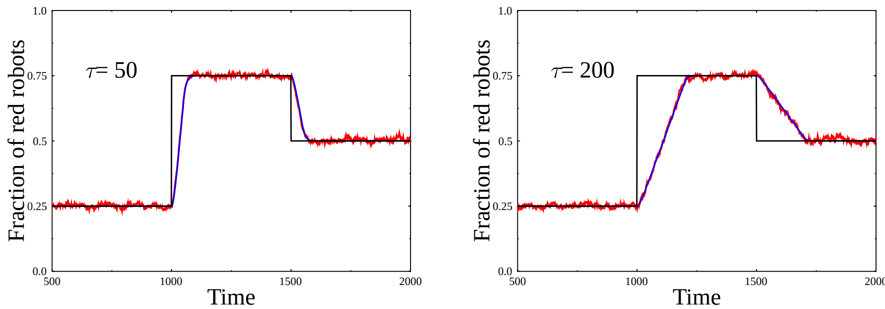

Fig. 3. Adaptation to changing puck distribution for different τ ($\varepsilon = 0.1$)

and then uses the transition rules described above to determine its new state, using the last τ entries in its registry. In Fig. 1 we plot the average fraction of red robots as a function of time for puck distribution $M_r = 500$, $M_g = 1500$, and for total number of robots $N = 100$, for different values of ε. We have averaged the plot over 100 trials. For comparison, we also plot $p_r(t)$ as given by Eq.12. One can see that the analytical curve fits perfectly with the results of the simulations. The fraction of robots in both cases converges to the same steady state value $p_0 = 0.25$, and the convergence time depends on ε as indicated by Eq.12.

The quality of performance in the task allocation scenario depends not only on the average number of robots collecting, say, red pucks, but also the fluctuations around this average. Hence, we studied the steady state probability distribution. Clearly, the strength of the fluctuations are characterized by the width of this distribution. To obtain the steady state probability distribution in the simulations, we used the time series generated by a single run. To avoid the effects of transient dynamics, we carried out simulations until the steady state was reached, and then constructed the histogram of $N_r(t)$–the number of red robots. The results are shown in Fig. 2 for different values of the fraction of red pucks. In each case, the distribution is peaked around its average value as one should expect. Again, one can see that there is an excellent agreement between the analytical curve (Eq. 20) and simulation results.

In Fig. 3 we plot the fraction of *Red* robots when the puck distribution undergoes step–like changes, both for simulations (averaged over 100 trials) and analytical results (Eqs. 14). One can see that the system adapts to the changes, and after some transient time the distribution of robots between the states reflects the puck distribution. Note that in this case also the analytical and simulation curves are virtually undistinguishable.

Finally, let us consider the case when $\alpha\tau M_0$ is sufficiently small so that the correction to the value of $\overline{\gamma}$ can not be neglected. As we mentioned above, in this

[2] Note that in our simulations we do not aim to reproduce realistic robot trajectories.

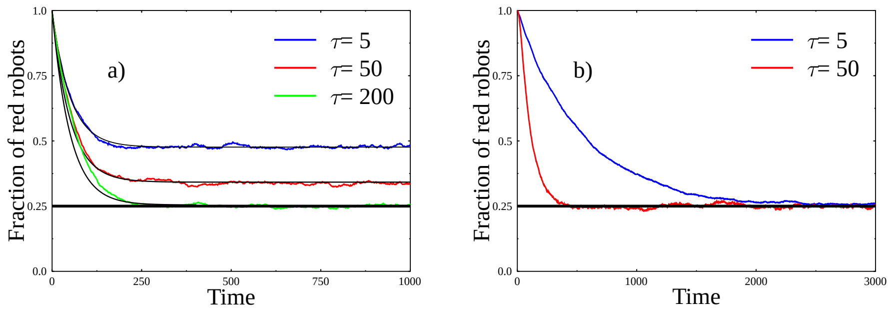

Fig. 4. a)Fraction of red robots vs time for different values of τ b)Fraction of red robots for modified transition rules. Both plots are the averages over 100 trials.

case steady state of Eq. 12 does not correspond to the puck distribution, $p_r^0 \neq \mu_0$, and in the limit $\alpha\tau M_0 \to 0$ the steady state converges to 1/2 not depending on μ_0. Note that this happens because for small enough $\alpha\tau M_0$ the robot's registry might not contain any readings at all. Hence, according to our rules,[3] each robot will choose either state with probability close to 1/2. This is illustrated in Fig. 4 (a) where we plot the number of red robots vs time for small overall density of pucks M_0/L^2 and different τ. Remarkably, the deviation from the desired steady state value is again well described by the analytical curve. Note also, that this undesired behavior can be avoided by modifying the transition rules as follows: if a robot's registry does not contain any reading for the last τ time steps, then the robot stays in its current state instead of choosing states with probability 1/2. This slight modification allows robots to achieve desired task allocation as shown in Fig. 4 (b).

8 Conclusion

In conclusion, we have presented a simple stochastic model of task allocation for multi–robot system, and studied it both analytically and in simulations. Dynamic task allocation model presented here is an adaptive form of foraging in a multi-robot system, where robots can switch dynamically between *Red* and *Green* foraging states. When a robot is in a *Red* foraging state, it is searching for and collecting *Red* pucks. The goal of dynamic task allocation is for the distribution of robots in *Red* and *Green* foraging states to dynamically adapt to the distribution of pucks, even when this distribution is not known in advance or changing in time. In order to accomplish this, robots make local observations of the pucks, estimate the density of each color based on past observations, and switch foraging state according to some transition function.

[3] Note that $\lim_{m_r \to 0} \lim_{m_g \to 0} \frac{m_r}{m_r + m_g} = 1/2$

We have studied this model analytically using annealed approximation of stochastic Master equation, where the robot's actual histories are replaced by random one drawn from Poisson distribution. Although it is not clear *a priori* that such an approximation is valid, we obtained excellent agreement with the results of numerical simulations. Note also that the model presented here can be generalized to the situations when there are more than two states for more general multi–agent settings.

The work presented in this paper does not address the role noise in observations caused by faulty robot sensors plays in the behavior of the system. Real robots making observations have crude video systems and may not be able to distinguish two objects that are overlapping in their visual field, or even their types (colors). Nor can robots uniquely identify objects or be able to tell whether the object they are seeing has been observed before. Such limitations will often lead robots to overestimate or underestimate environmental states, and will require further elaboration of the analytical techniques described here. Capturing noisy observations and studying their effect on the collective behavior of an adaptive system is the focus of our ongoing research.

Acknowledgment

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-00-2-0573. The authors would like to thank Chris Jones for introducing them to his task allocation system and useful discussions of the system behavior.

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, October 2002, Hammamet, Tunisia*. IEEE Press, 2002.
2. Dani Goldberg and Maja J. Matarić. Coordinating mobile robot group behavior using a model of interaction dynamics. In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 100–107, Seattle, WA, USA, 1999. ACM Press.
3. 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).
4. 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.
5. Chris V. Jones and Maja J Matarić. Adaptive task allocation in large-scale multi-robot systems. In *Proceedings of the 2003 (ICRA'03), Las Vegas, NV*. IEEE, 2003.
6. Sanza Kazadi, A. Abdul-Khaliq, and Ron Goodman. On the convergence of puck clustering systems. *Robotics and Autonomous Systems*, 38(2):93–117, 2002.
7. Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.

8. Kristina Lerman and Aram Galstyan. Macroscopic Analysis of Adaptive Task Allocation in Robots. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS-2003), Las Vegas, NV*, Oct 2003.
9. 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.
10. Ling Li, Alcherio Martinoli., and Yasser Abu-Mostafa. *Emergent Specialization in Swarm Systems*, volume 2412 of *Lecture Notes in Computer Science*, pages 261–266. Springer Verlag, New York, NY, 2002.
11. 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.
12. Alcherio Martinoli and Kjerstin Easton. Modeling swarm robotic systems. In B. Siciliano and P. Dario, editors, *Proc. of the Eight Int. Symp. on Experimental Robotics ISER-02, Sant'Angelo d'Ischia, Italy*, Springer Tracts in Advanced Robotics 5, pages 297–306, New York, NY, July 2003. Springer Verlag.
13. M. J. Matarić. Learning in behavior-based multi-robot systems: Policies, models, and other agents. *Cognitive Systems Research*, 2(1):81–93, Apr 2001.
14. Martin Riedmiller and Arthur Merke. Karlsruhe brainstormers - a reinforcement learning approach to robotic soccer II. In *RoboCup-01: Robot Soccer World Cup V*, LNCS. Springer, 2001.
15. Peter Stone and Richard S. Sutton. Scaling reinforcement learning toward RoboCup soccer. In *Proc. 18th International Conf. on Machine Learning*, pages 537–544. Morgan Kaufmann, San Francisco, CA, 2001.
16. Ken Sugawara and Masaki Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica*, D100:343–354, 1997.
17. Ken Sugawara, Masaki Sano, Ikuo Yoshihara, and K. Abe. Cooperative behavior of interacting robots. *Artificial Life and Robotics*, 2:62–67, 1998.
18. N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier Science, 1992.