Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Building Redundancy in Multi-Agent Systems Using Probabilistic Action

Annie S. Wu, R. Paul Wiegand, and Ramya Pradhan
University of Central Florida
Orlando, FL 32816
aswu@cs.ucf.edu, wiegand@ist.ucf.edu, ramya.pradhan@knights.ucf.edu

Abstract

In this paper, we examine the effects of probabilistic response on a task allocation problem for a decentralized multi-agent system (MAS) and how such a mechanism may be used to tune the level of redundancy in an MAS. Redundancy refers to a back up pool of agents, beyond the necessary number required to act on a task, that have experience on that task. We present a formal analysis of a response threshold based system in which agents act probabilistically and show that we can estimate the response probability value needed to ensure that a given number of agents will act and that we can estimate the response probability value needed to achieve a given level of redundancy in the system. We perform an empirical study using an agent-based simulation to verify expectations from the formal analysis.

Introduction

In this paper, we study the use of probabilistic response in a decentralized multi-agent system (MAS) and examine how such a mechanism may be used to adjust and adapt the amount of redundancy in an MAS. One of the expected advantages of multi-agent systems over single agent systems is robustness due to redundancy. If one agent in an MAS is unable to perform its task, other agents may be available to fill in. For tasks where experience improves performance, having a "backup pool" of agents with some experience on a task can mitigate a significant drop in performance should the primary actor or actors, those agents that are primarily responsible for a task, become unexpectedly unavailable or lost. Thus, on tasks where experience affects performance, an MAS will need to balance having the most experienced agents perform a task efficiently with giving inexperienced and, consequently, less efficient agents a chance to gain experience on the task. For example, in a search and rescue mission, experienced agents are likely to be more effective at finding and extracting victims than inexperienced agents; however, inexperienced agents need to be given the opportunity to perform the task in order to gain experience and improve their performance on the task. We show that a probabilistic response mechanism may be used to dynamically assemble and maintain a backup pool of partially experienced agents in a decentralized system.

In centralized systems, forming and maintaining a backup pool of agents for a given task is relatively straightforward. Given the desired size of the backup pool, a central controller can explicitly balance the exploitation of experienced agents with the training of backup members. An example strategy would be to occassionally assign the task to backup members to build their experience while concentrating most assignments on primary actors to maintain performance. In addition, when primary actors for a task are lost, backup members can be promoted, and new backup members are trained from as yet inexperienced agents. Centralized systems, however, are vulnerable due to their single point of failure and often do not scale well as the team size increases.

In decentralized systems where agents act independently and there is no central controller, ensuring that the appropriate agents act (primary actors most of the time and backup agents occassionally to gain experience) is more challenging. Although it may be possible for all agents to negotiate and come to a consensus on who takes what task, such methods that require non-trivial communication and computation do not scale well. If reliable communication and global knowledge are not guaranteed, then agents must decide when to take on a task with little to no information about what their team members are doing.

A common approach to task allocation in decentralized MAS is the response threshold approach. In this approach, agents have a threshold for each task and consider acting on a task when the task stimulus exceeds their threshold for the task. If agents have different thresholds for the same task, those agents with lower thresholds act sooner than those with higher thresholds, and those with higher thresholds may not need to act at all if the earlier actors sufficiently address the task's needs. As a result, some coordination among agents is possible even when there is little or no communication between agents.

The response threshold model was originally proposed as a mathematical model of division of labor in social insect societies (Bonabeau, Theraulaz, and Deneubourg 1996; 1998). A key element of the biological model is that agents' responses to task stimuli exceeding their thresholds are probabilistic rather than deterministic. A probabilistic response means that an agent is not guaranteed to act when a task stimulus exceeds its threshold. This is thought to increase the robustness of insect societies by allowing agents with higher thresholds for a task occassional opportunities to act and gain experience on the task, in effect forming a backup pool (Weidenmüller 2004).

The response threshold approach has been used in multiple computational domains involving coordination of decentralized MAS (Agassounon and Martinoli 2002; Campos et al. 2000; Krieger and Billeter 2000; Nouyan 2002; Parker 1998; Price and Tino 2004). Both deterministic and probabilistic approaches have been shown to be capable of decentralized task allocation. To our knowledge, however, there is not yet a thorough study examining the effects of probabilistic action on building a back-up pool of experienced agents — and, in particular, no formal analysis of such a process has been conducted.

We show that, for decentralized MAS that use response thresholds to determine task allocation among agents, a simple technique based on probabilistic action can be used to generate a backup pool of agents with some experience for a task while keeping most of the work on the most experienced agents to maintain performance. Probabilistic action is implemented via a *response probability*, which is defined to be the probability that an agent will act on a task when its action threshold is reached. We present a formal analysis of this response probability model and show how it affects the ability of an MAS to satisfy a task's needs and to form a backup pool. In addition, we show that the response probability may be varied to adjust the size of the backup pool.

Our model

Our model focuses on a task allocation problem with a single task that needs to be addressed. Given an MAS consisting of n agents, each time the task occurs (needs to be addressed), $x : x < n$ agents from the MAS must respond in order to satisfy the task. Since our MAS is decentralized, each agent independently decides whether or not to respond to a task. If x agents act in response to a task, we are successful in forming a response team for the task. If fewer than x agents act on a task, we do not have enough actors to meet the task's needs and we are unsuccessful in forming a response team.

Our MAS is a response threshold system in which each agent has a threshold value for each task. If we order the agents by increasing task threshold from left to right, those agents at the left end of the line are the first to act and have more chances to gain experience on the task, and those agents at the right end of the line are the last to act and have few to no chances to gain experience on the task.

To make the model probabilistic, we introduce a response probability, $s : 0.0 \leq s \leq 1.0$, that affects whether or not an agent will act once its threshold has been met. If an agent's threshold is met before the task is satisfied (before x agents have responded to the task), that agent becomes a *candidate* and is offered an opportunity to act. A candidate chooses to act with probability s. If the candidate does choose to act, it becomes an *actor*, and the number of responders to the task increases by one; if it does not choose to act, then the next agent in the ordering becomes a candidate. Thus, the response probability is the probability that a candidate will become an actor. As we move from left to right in the ordered list of agents (moving from fastest to slowest responder), the addition of the response probability s means that the x responders that satisfy the task may not always be the left most x agents. If some of those agents probabilistically choose not to act, then agents beyond the first x can receive an opportunity to act and gain experience.

Figure 1 illustrates the effects of s. We define a *trial* to be one instance in which a task requires action from the MAS. When $s = 1.0$, the system is deterministic and only the first x agents gain experience on the task regardless of the number of trials. When $0.0 < s < 1.0$, agents act probabilistically. While only a maximum of x agents gain experience in each trial (though not necessarily the first x agents), over multiple trials, we expect more than the first x agents to gain experience on the task.

Because x actors are required to satisfy the task, the first x agents in the ordered list will always become candidates, which means that they will get an opportunity to gain experience for every instance of the task. Agents beyond the first x in the ordering will only become candidates if a full response team has not been achieved by the time their turn comes up. The *pool* is the set of agents beyond the first x agents that have gained experience over multiple trials. At high s values, we expect the pool size to be close to zero because most of the first x agents have a high probability of choosing to act. As the s value decreases, the pool size will increase because the first x agents, as well as agents beyond the first x agents, have a decreasing probability of choosing to act. As we approach $s \leq x/n$, however, decreasing s will also increase the chance that the task will not be satisfied because it will be increasingly likely that fewer than x out of all n possible agents choose to act.

Thus, we see that s affects both the ability of an MAS to form a response team and the level of redundancy that can be achieved by the MAS over multiple trials. The analysis that follows seeks to find guidance on how to set s for an MAS, depending on whether the goals are to successfully form a response team, to maximize the number of agents that gain experience on the task, or to meet specific redundancy requirements.

Analysis

Our analysis examines two aspects of the response probability, s. First, how does s affect the ability of an MAS to successfully form a response team. Second, how does s affect the ability of an MAS to build and maintain redundancy.

The reader will recall traditional asymptotic notation. We use $O(g(n))$ to denote the set of functions that asymptotically bound $g(n)$ from above and $\Omega(g(n))$ to denote the set of functions that asymptotically bound $g(n)$ from below. Formally:

$$O(g(n)) = \left\{ \begin{array}{l} f(n) : \text{there exist positive constants } c \\ \text{and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \\ \text{for all } n \geq n_0 \end{array} \right\}$$

$$\Omega(g(n)) = \left\{ \begin{array}{l} f(n) : \text{there exist positive constants } c \\ \text{and } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \\ \text{for all } n \geq n_0 \end{array} \right\}$$

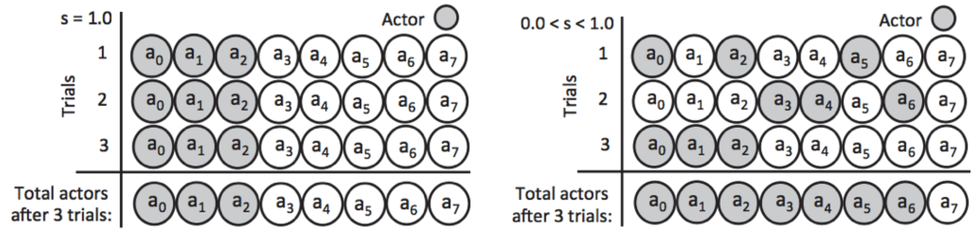

Figure 1: Example of the impact of response probability s. The total number of agents is $n = 8$ and the number of agents required to perform the task is $x = 3$. The number of trials is three. When $s = 1.0$, only the first three agents become candidates and then actors. No additional actors gain experience so there is no back up pool. When $0.0 < s < 1.0$, the first three agents always become candidates and stochastically may or may not become actors based on s. In this example, $s = 0.5$ and a coin toss was used to determine if a candidate become an actor. Over three trials, the total number of agents that gain experience is seven, giving a back up pool of four extra agents with experience.

For those less familiar with this notation, it may be helpful for our discussion to point out that the class of functions $e^{-\Omega(g(n))}$ is the set of functions that drop toward zero as an exponential function of $g(n)$ — or faster.

Forming a team

Recall that an MAS is only successful in responding to a task if it can form a response team, and a team is formed when x agents choose to work on the task. While we can ensure that all agents in an MAS will gain experience on a task over multiple task instances by simply setting s to a low value, having an MAS full of "experienced" agents is pointless if the MAS is unable to form a full response team. As a result, the first question that we would like to ask is, given that we have n agents and a task that requires $x : x < n$ agents, can we determine what s values will allow the MAS to successfully form a reponse team?

We begin by traversing all n agents and marking each agent with probability s. This mark indicates the probability that an agent will choose to act should it become a candidate. Let M be a random variable specifying the total number of marked agents, regardless of whether the agents participate in the team or not.

Lemma 1 *A single trial of the task allocation process will result in $M \leq x - 1$ with probability $1 - e^{-\Omega(n)}$ when $s < \frac{x-1}{en}$. It will result in $M \geq x$ with probability $1 - e^{-\Omega(n)}$ when $s > \frac{x}{n}$.*

Proof: *The expected number of marked agents is $n \cdot s$, $E\{M\} = ns$, since there are n agents to be marked, and each is marked with independent probability s. Let*

$$\delta = \frac{x-1}{ns} - 1$$

and note that

$$(1+\delta)\, ns = \left(1 + \frac{x-1}{ns} - 1\right) ns = x - 1$$

We use Chernoff inequality to bound the probability that there are more than $x - 1$ marks.

$$\begin{aligned} Pr\{M > x-1\} &= Pr\{M > (1+\delta)E\{M\}\} \\ &< \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{E\{M\}} \\ &= \left[\frac{e^{\frac{x-1}{ns}-1}}{\left(\frac{x-1}{ns}\right)^{\frac{x-1}{ns}}}\right]^{ns} \\ &= \frac{1}{e^{ns}}\left[\left(\frac{ens}{x-1}\right)^{\frac{x-1}{ns}}\right]^{ns} \end{aligned}$$

If $x - 1 > ens$, this converges to 0 exponentially fast as n grows. Thus $s < \frac{x-1}{en}$ implies $Pr\{M < x\} = 1 - e^{-\Omega(n)}$.

Now consider $\delta = 1 - \frac{x}{ns}$ and note that

$$(1-\delta)\, ns = \left(1 - 1 + \frac{x}{ns}\right) ns = x$$

We use Chernoff inequality to bound the probability that there are fewer than x marks.

$$\begin{aligned} Pr\{M < x\} &= Pr\{M < (1-\delta)E\{M\}\} \\ &< e^{\delta^2 E\{M\}/2} \\ &= e^{\left(1-\frac{x}{ns}\right)^2 ns/2} \\ &= e^{-\frac{ns}{2}(ns-x)^2} \end{aligned}$$

If $x < ns$, this converges to 0 exponentially fast as n grows. Thus $s > \frac{x}{n}$ implies $Pr\{M \geq x\} = 1 - e^{-\Omega(n)}$. □

Theorem 1 *With high probability, as n grows a complete team will almost surely be formed when $s > \frac{x}{n}$ and will almost surely not be formed when $s < \frac{x-1}{en}$.*

Proof: *If there are fewer than x marks over all n agents, a complete team of x agents will not be formed, and a complete team can only be formed if there are x or more marks. Noting this, the conclusion follows from Lemma 1.* □

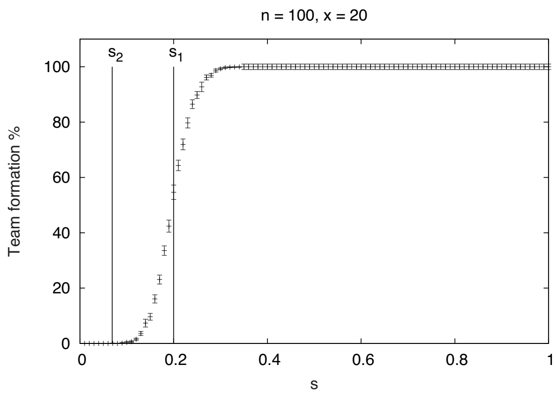

Figure 2: Comparison of calculated values for s_1 and s_2 with empirical data on the percentage of trials in a run in which a team is formed, averaged over 20 runs, as s varies from 0.0 to 1.0. $s_1 = 0.2$ is the response probability value above which a team is likely to be formed in a single trial and $s_2 \approx 0.07$ is the response probability value below which a team is unlikely to be formed in a single trial.

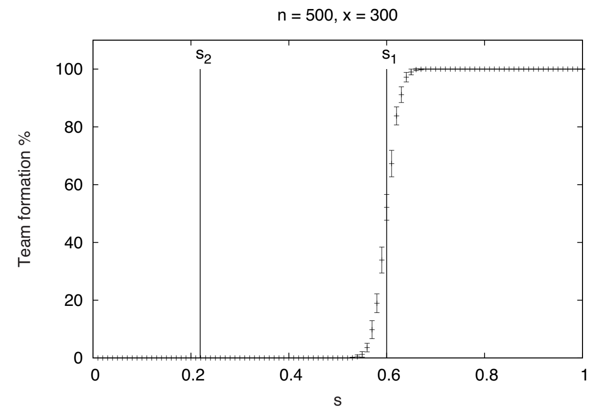

Figure 3: Comparison of calculated values for s_1 and s_2 with empirical data on the percentage of trials in a run in which a team is formed, averaged over 20 runs, as s varies from 0.0 to 1.0. $s_1 = 0.6$ is the response probability value above which a team is likely to be formed in a single trial and $s_2 \approx 0.22$ is the response probability value below which a team is unlikely to be formed in a single trial.

Asymptotic analysis gives us bounds for the response probability as n grows; however, it is constructive to see how this practically relates to a specific scenario. Consider a system with $n = 100$ agents and a task that requires $x = 20$ agents. If s_1 equals the value of s for which a team is likely to be formed in a single trial, then

$$s_1 > \frac{x}{n} = \frac{20}{100} = 0.2$$

If s_2 equals the value of s for which a team is almost surely not be formed in a single trial, then

$$s_2 < \frac{x-1}{en} = \frac{19}{2.71828183 * 100} = 0.069897 \approx 0.07$$

Figures 2 and 3 compare the predicted values for s_1 and s_2 with team formation data from an agent-based simulation. In the simulation, each agent has a threshold for the task; lower thresholds indicate quicker response. Recall that one trial is defined to be one instance of the task. In each trial, agents, in order of increasing thresholds, decide whether or not to act with a probability s. The trial ends when either a response team is formed or when all agents have been offered the opportunity to act on the task. Each *run* of the simulation consists of 100 trials and we record the number of trials out of 100 in which a team is formed.

Figure 2 shows data from a simulation consisting of 100 agents ($n = 100$) and a task that requires 20 agents ($x = 20$). Figure 3 shows data from a simulation consisting of 500 agents ($n = 500$) and a task that requires 300 agents ($x = 300$). Simulations using other n and x values yield similar results. The x-axis in both figures plots the s values where $0.01 \leq s \leq 1.0$. The y-axis shows the percentage of trials in a run in which a team is formed, averaged over 20 runs. The s_1 and s_2 values calculated above are indicated by two vertical lines. In both examples, the calculated value for s_1, which indicates the threshold above which a team is likely to be formed in a single trial, tends to fall on s values where the team formation percentage is fifty percent or above. The calculated value for s_2, which indicates the threshold below which a team is unlikely to be formed in a single trial, falls on s values where where team formation percentage is zero percent. It is likely that the $\frac{x}{n}$ bound is tight and that $\frac{x-1}{en}$ is overly cautious.

Building and maintaining redundancy

Redundancy is achieved when there is a pool of agents beyond the first x agents that have gained experience on the task. Because, in each trial, a maximum of x agents can act and gain experience, redundancy can only be generated over multiple trials. The second question we ask is whether we can determine the appropriate value of s to use in a system when a specified level of redundancy is desired. To do so, we first look at what happens in a single trial and ignore the possibility of failing to make a team, then examine the combination of these results with the recommendations on team formation from the previous section.

We define c to be the desired level of redundancy where cx is the desired number of agents with experience (and the size of the pool is $cx - x$). Our goal is to determine the s values for which the cx^{th} agent is highly likely to have gained experience and be part of the pool. The probability of the i^{th} agent acting and therefore gaining experience (P_i) is no smaller for agents preceding the cx^{th} in the ordering (i.e., $P_i \geq P_{cx}$ if $i \leq cx$) since those will be given the opportunity to act sooner and all agents have the same s. Here we show formally that if $s \leq \frac{1}{ec}$, there is a constant probability that the cx^{th} agent will gain experience and if $s > \frac{1}{c}$ then it will almost certainly fail to gain experience. After deriving these bounds, we present an empirical case to show that the real system is consistent with our formal advice.

The proof for this follows a similar structure as the above proof, and we begin by traversing all n agents and marking each agent with probability s. Let K be a random variable specifying the number of marks in the first cx agents. We first describe a bound on s that is *sufficient* to assure a reasonable P_i, then we describe a looser bound that is *required* if we do not want P_i to converge to 0 with team size.

Theorem 2 *In a single trial of the task allocation process, if* $s \leq \frac{1}{ec}$, *then* $P_{cx} = 1 - e^{-\Omega(1)}$.

Proof: *The expected number of marks in the first cx agents is* $c \cdot x \cdot s$, $E\{K\} = cxs$. *Let*

$$\delta = \frac{1}{cs} - 1$$

and note that

$$(1+\delta)\, cxs = \left(1 + \frac{1}{cs} - 1\right) cxs = x$$

We use Chernoff inequality to bound the probability that there are at least x marks in the first cx agents:

$$\begin{aligned} Pr\{K \geq x\} &= Pr\{K > (1+\delta)E\{K\}\} \\ &< \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{E\{K\}} \\ &= e^{-csx} \cdot (ecs)^x \end{aligned}$$

So when $s < \frac{1}{ec}$, *this approaches 0 exponentially fast with* x. *Thus the* cx^{th} *agent almost surely is given the opportunity to act and* $P_{cx} = s \cdot \left(1 - e^{-\Omega(x)}\right) \approx s = 1 - e^{-\Omega(1)}$.

Theorem 2 gives us a bound for *sufficient* values of s for the cx^{th} agent to gain experience. If the cx^{th} agent has a constant probability of gaining experience in a single trial, then a constant number of repeated trials will ensure the agent eventually gains experience. As stated, agents that precede the cx^{th} agent in the ordering will have no worse probability of gaining experience, so the same logic applies to all of them. The question of how many trials are necessary to ensure this experience is a different one which is addressed by (Wu et al. 2012). Now that we have seen what sufficient values of s are to ensure experience, let us examine what values of s are so large that they prevent the cx^{th} agent from even being given the opportunity to gain experience.

Theorem 3 *In a single trial of the task allocation process, if* $s > \frac{1}{c}$, *then* $P_{cx} = e^{-\Omega(x)}$.

Proof: *Again* $E\{K\} = cxs$. *Now let*

$$\delta = 1 - \frac{1}{cs}$$

and note that

$$(1-\delta)\, cxs = \left(1 - 1 + \frac{1}{cs}\right) cxs = x$$

We use Chernoff inequality to bound the probability that there are fewer than x marks in the first cx agents:

$$\begin{aligned} Pr\{K < x\} &= Pr\{K < (1-\delta)E\{K\}\} \\ &< e^{\delta^2 E\{K\}/2} \\ &= e^{-\frac{x}{2cs}\left(1-\frac{1}{cs}\right)^2} \end{aligned}$$

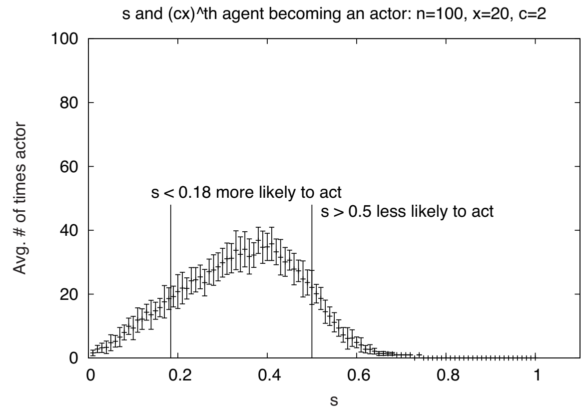

Figure 4: The influence of s on the cx^{th} agent becoming an actor when $c = 2$. The vertical lines are the formal bounds.

So when $s > \frac{1}{c}$, *this approaches 0 exponentially fast with* x. *Thus the probability that the* cx^{th} *agent is even given the opportunity to act is exponentially small for constant* s. □

Theorem 3 tells us that when $s > \frac{1}{c}$, later agents will very probably be starved of the opportunity to gain experience by the agents before them in the ordering. This expectation means that unless there are an exponential number of trials, it is unlikely that repeated trials will allow us to build up a redundancy of cx agents with experience. Of course, these are asymptotic results that become more correct as team size and redundancy factor increase. Therefore, it is instructive to consider empirical results for specific values.

Consider again the scenario in which there are 100 agents in all ($n = 100$) and we face a task that requires 20 agents ($x = 20$). We would like a redundancy factor of 2 ($c = 2$), which means that we are hoping that the first 40 agents will gain experience over time on that task given enough trials.

Figure 4 shows the influence of s on the cx^{th} agent becoming an actor. The x-axis plots s values. The y-axis measures the average and standard deviation of the number of times the cx^{th} (here the 40^{th}) agent becomes an actor in 20 simulations of 100 trials each. From our theory, we see that the 40^{th} agent is more likely to act when $s < \frac{1}{ec} \approx 0.18$, and less likely to act when $s > \frac{1}{c} = 0.5$. Starting at $s = 0.18$, the plot shows an increase in the number of times this agent becomes an actor as the s value increases. As s increases beyond $s > 0.5$, the plot shows a decrease in the number of times this agent becomes an actor with increasing s value, to the point that it no longer acts at $s \approx 0.75$.

The results shown in Figure 4 demonstrate that the number of times the cx^{th} agent acts is consistent with our formal, probabilistic predictions; however, it does not necessarily confirm directly that redundancy within the system is consistent with our theory. Figure 5 shows the influence of s on the formation of backup pool of agents. Again, the x-axis plots s values, but now the y-axis measures the average number of actors in the 20 simulations. Figure 5 shows that, for the s value when the cx^{th} agent is likely to act ($s \approx 0.18$), the maximum average number of actors obtained is around

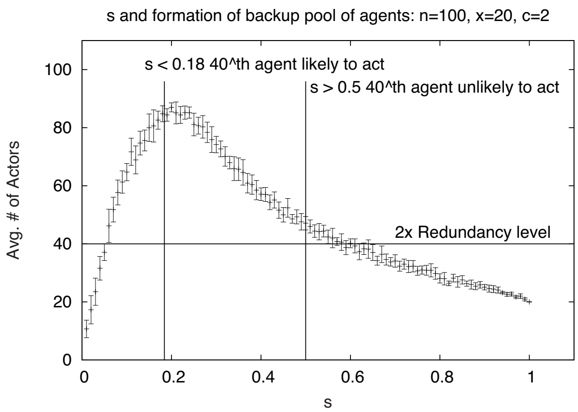

Figure 5: Effect of s on the formation of backup pool of $c = 2$ in a single task. The vertical lines indicate our formal bounds, and the horizontal line shows when the redundancy factor is met.

85, which means the cx^{th} or 40^{th} agent has acted and is in the backup pool and the desired redundancy is achieved. At the s value above which the cx^{th} agent is unlikely to act ($s = 0.5$), the maximum number of actors obtained is around 48. Again, the cx^{th} or 40^{th} agent makes it into the backup pool and the desired redundancy is achieved. These empirical results are consistent with our formal predictions even though the values for n, x, and c are relatively small. Moreover, for large c, the difference between $\frac{1}{c}$ and $\frac{1}{ec}$ become smaller, making the advice more specific as more redundancy is required.

Astute readers will note that the average number of actors again drops when s is too small, even though $s < \frac{1}{ec}$. This occurs for a different reason: complete teams are not being formed when s is too small. Combining the theoretical advice from the previous section with this section and assuming the user wants to maximize the back up pool with *at least* cx agents with experience, we see that our theory can offer very specific and tangible advice.

Referring back to Figure 2 where $n = 100$ and $x = 20$, complete teams will almost certainly form if $s \geq \frac{x}{n} = 0.2$ and, according to Figure 5, c-factor redundancy is ensured if $s < \frac{1}{ec} \approx 0.18$. As we can see in Figure 5, an s value in the range $0.18 < s < 0.2$ is indeed a good first choice to achieve c=2 redundancy. Note that if $s < \frac{x-1}{en} \approx 0.07$, we will almost certainly *fail* to form a team — which is roughly close to where we lose a redundancy factor of two on the left part of the graph. Also, if $s > \frac{1}{c} = 0.5$, we will almost certainly starve later agents and fail to get redundancy, which is roughly close to where we lose a redundancy factor of two on the right part of the graph.

Conclusions

In this work, we examine the use of an agent response probability as a method for generating redundancy in an MAS. Redundancy is defined as a pool of extra agents with experience on a task beyond the necessary number of agents for that task. Having agents respond probabilistically rather than deterministically to a task means that not all of the primary actors for a task may act when a task requires attention, giving other agents an opportunity to act and gain experience on the task.

We present a formal analysis of a response threshold based system in which agents act probabilistically. Given the number of agents in the multi-agent system and the number of agents needed to satisfiy a task, we are able to estimate two factors regarding the system's response probability value. First, we can estimate the range of response probability values that are likely to allow the MAS to satisfy the task demands. Second, we can estimate the range of response probability values that are likely to allow the MAS to achieve a specified level of redundancy. Empirical studies using an agent-based simulation of the problem support our theoretical results.

This work shows that a simple response probability model may be used in decentralized multi-agent systems to dynamically build and maintain redundancy in terms of agent experience. We are able to model such a system theoretically and use this model to provide guidance on how to choose response probability values for a given problem and goals.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported in part by ONR grant #N000140911043.

References

Agassounon, W., and Martinoli, A. 2002. Efficience and robustness of theshold-based distributed allocation algorithms in multi-agent systems. In *Proc. Int. Joint Conf. Auto. Agents & MAS*, 1090–1097.

Bonabeau, E.; Theraulaz, G.; and Deneubourg, J. 1996. Quantitative study of the fixed response threshold model for the regulation of division of labor in insect societies. *Biological Sciences* 362(1376):1565–1569.

Bonabeau, E.; Theraulaz, G.; and Deneubourg, J. 1998. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Math. Biol.* 60:753–807.

Campos, M.; Bonabeau, E.; Theraulaz, G.; and Deneubourg, J. 2000. Dynamic scheduling and division of labor in social insects. *Adaptive Behavior* 8(2):83–96.

Krieger, M. J. B., and Billeter, J.-B. 2000. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems* 30:65–84.

Nouyan, S. 2002. Agent-based approach to dynamic task allocation. In *Proc. ANTS, LNCS 2463*, 28–39.

Parker, L. E. 1998. ALLIANCE: An architecture for fault tolerant multi-robot cooperation. *IEEE Trans. Robot. & Automation* 14(2).

Price, R., and Tino, P. 2004. Evaluation of adaptive nature inspired task allocation against alternate decentralised multiagent strategies. In *Proc. PPSN VIII, LNCS 3242*, 982–990.

Weidenmüller, A. 2004. The control of nest climate in bumblebee (*bombus terrestris*) colonies: interindividual variability and self reinforcement in fanning response. *Behav. Ecology* 15(1):120–128.

Wu, A. S.; Wiegand, R. P.; Pradhan, R.; and Anil, G. 2012. The effects of inter-agent variation on developing stable and robust teams. In *Proceedings of the AAAI Spring Symposium*.