On the relationship between response probability and redundancy in teams of collaborating agents

Ramya Pradhan and Annie S. Wu
Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362
ramya.pradhan@knights.ucf.edu, aswu@cs.ucf.edu

Abstract—One of the expected benefits of multi-agent systems is their capacity for redundancy which is expected to promote system robustness. We examine the impact agents' probabilistic action has on generating redundancy in multi-agent systems with no direct inter-agent communication. We present a mathematical model of such a system, analyze the factors that influence it, and empirically examine its effects on multi-agent systems under perturbations such as removal and insertion of agents. Results indicate that agents' probabilistic action can allow a team of collaborating agents to build a pool of agents with experience and this redundant pool can promote system robustness under perturbations.

Index Terms—Response probability, Multi-agent systems, Robustness, Redundancy, Mathematical framework

I. INTRODUCTION

A multi-agent system (MAS) is a human engineered system consisting of multiple agents that work together to accomplish a common goal or an objective. Collaboration among agents in a MAS to accomplish this goal requires coordination among agents to take on all necessary and appropriate roles. In the process of collaboration, agents need to be able to recognize if roles are unfilled and be able to take on such unfilled roles. In situations where experience improves agent performance on a task, it is beneficial to have a pool of extra agents with experience on each task.

As with any human engineered system, one of the desired system features is its ability to be robust with some form of a back up mechanism consisting of identical redundant components that are capable of taking up tasks of a failed component. Thus, the ability to build a backup pool of agents for each role is important in promoting robustness of a team of collaborating agents. One of the expected benefits of MAS's is their inherent potential for redundancy [1], [2]. If an agent is disabled or lost, there exists a potential that another agent can take over the role of the missing agent.

We examine how a stochastic tendency of an agent to act, as defined by a response probability, can generate redundancy and improve collaboration in MAS's with no direct interagent communication. Response probability refers to the probability that an agent will respond or act when its response threshold has been met [3]. The fact that this response is probabilistic means that, unlike traditional response threshold systems, an agent may not be guaranteed to act when its response threshold is met. As a result, the lower threshold or the most willing agents will not always act on a task thus giving higher threshold agents an opportunity to act on the task. Over multiple instances of a task, this potentially generates a larger pool of experienced agents, effectively generating redundancy.

Redundancy in engineering as defined by the Oxford dictionary is "the inclusion of extra components which are not strictly necessary to functioning, in case of failure in other components". Accordingly, redundancy is expected to promote a system's stability against potential component failure that could lead to system failure. An attractive feature of the MAS architecture for designers of engineering applications is its ability to naturally lend itself to redundancy due to the presence of many agents. This feature has a potential of increasing the robustness of such engineering applications [4]. Studies have shown that the existence of redundancy in an MAS can promote system stability [1], [2].

The simplest way to introduce redundancy into a system is to add extra agents for each task to the system. The robustness of this approach, however, is bounded by the number of extra agents added and the system may not be able to adapt if all redundant copies of a type of agent is lost. Threshold-based systems provide additional flexibility in that potentially a large number, if not all, of the agents may have the capability to act on a given task. Agents have a threshold for each task and act when a task stimulus reaches its threshold for that task. If low threshold agents are lost and do not perform a task, higher threshold agents' thresholds will eventually be reached and they will act. Although the threshold based approach provides greater adaptability in terms of having "back up" agents available for a given task, it typically does not actively generate a large experience pool. The low threshold agents will always be the first to act and gain experience, and the high threshold agents will not act unless there are not enough low threshold agents. We believe that a simple addition of a response probability, a nondeterministic probability with which an agent will act once its threshold has been reached, can provide an elegant method to generate redundancy in response threshold systems.

In this paper, we examine the relationship between agents' response probability and system redundancy. We mathematically model a system with agents that exhibit this behavior in a single task environment with no direct interagent communication. We empirically study its effects on system

redundancy and performance under insertion and removal of agents. Our results show that our model can effectively predict relationship between response probability and redundancy in empirical systems.

II. An overview of response probability usage variations in multi-agent systems

The term *response probability* has been used before in the multi-agent task allocation literature. Examination of the literature reveals that this term is used in two different contexts. We categorize these two contexts as deterministic [5]–[9] and stochastic [10]–[12]. We present a broad overview of the variations in using response probability for task allocation in multi-agent systems in relation to our study.

A. Response probability and deterministic decision making

Response probability when used in the *deterministic* sense refers to a description of an agent's preference for a task relative to its preference for other tasks. This value can change over time and is affected by what tasks an agent acts on. Agents use the response probability value to choose among available tasks. Given one or more tasks, an agent will choose one with the highest response probability and will always act on that task. For example, if an agent's response probability for $task$ 1 is 0.3 and for $task$ 2 is 0.1, then it will choose and act on $task$ 1. It is not the case that it acts on $task$ 1 with a 30% probability. As such there is not actually a stochastic element in the agent's decision to choose or act on a task.

This deterministic process of decision making was first utilized by Bonabeau et. al [5]. Their study uses a response function to calculate agents' response probability values. A variation of this response function where the response threshold includes weighted parameters, such as remembering and forgetting coefficients, has been discussed by Campos et. al [6], Dorigo et. al [7], Reijers et. al [8], and Yu and Ram [9].

B. Response probability and stochastic decision making

Response probability when used in the *stochastic* sense refers to the probability that an agent will act on a given task once that task has been assigned to it. Agents use the response probability value to decide if they are going to act or not on the assigned task. For the same example of an agent's response probability of 0.3 for $task$ 1 and 0.1 for $task$ 2, if the agent is assigned $task$ 1, it will act on the task with a 30% probability and similarly, with a 10% probability on $task$ 2. Therefore, there actually exists a stochastic element in the agent's decision to act on a task.

This stochastic process of decision making has been discussed and utilized by Merkle et. al [10] in a worker-helper scenario where a helper acts on a given task based on its response probability. Similar worker-helper scenarios are found in Xiang and Lee [11] and Scheidler et. al. [12]. Price and Tino [13] use response probability as an alternate strategy to the variable response threshold model for task allocation in a mail retrieval system where every agent retrieves mail based on its response probability value.

III. Problem Definition

We study an MAS consisting of n agents and a single task that requires x agents to act, where $x \leq n$. Every agent has a threshold for working on the task. Thresholds enable agents to collaborate in a task environment with no direct interagent communication. If an agent has a low threshold value for the task, then it is quicker to act on the task. Conversely, if an agent has a high threshold value for the task, then it is slower to act on the task. This variation in how quickly an agent will act on the task defines an ordering in the agent set. Accordingly, let agent a_0 be the first and a_{n-1} be the last to respond to the task as shown in Figure 1. Let $prob\_act$ be the response probability with which agents will act once their response threshold is met.
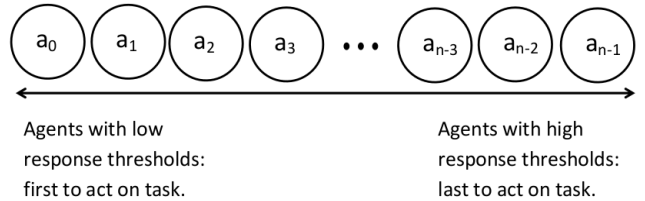

Fig. 1. Agents ordered by their response threshold for a task. Agent a_0 is most likely to act on the task due to its low threshold value for performing it and agent a_{n-1} is least likely to act on it due to its high threshold value for performing it.

Agents can be in one of three states at any given time: *Inactive*, *Candidate*, or *Actor*. We define these states as
1) $Inactive$: An agent that is not a candidate or an actor.
2) $Candidate$: An agent that is given the opportunity to act on the task.
3) $Actor$: An agent that acts on the task.

Figure 2 describes the steps that make up a single trial of our system. A single $trial$ simulates one instance in which the task needs to be addressed by the MAS and offers one opportunity for agents to gain experience. Let i count agents in order of their threshold values and let x_a count the number of agents that have acted so far. All agents begin in the inactive state. Agents in order of their willingness to act become candidates while $x_a \leq x$. Let C_i be the probability with which an agent becomes candidate in a single trial. Let P_i be the probability with which a candidate becomes an actor. If an agent becomes an actor in a single trial, that agent gains experience on the task. A trial ends when x agents have become actors or when all agents have had the opportunity to become actors. Within a single trial, a maximum of x agents will act and gain experience. Over multiple trials, potentially more than x agents may gain experience if not all of the same agents act in each trial.

We develop a mathematical model that explores the relation between the system response probability and agents' ability to gain experience over multiple trials. This model is used to answer the following questions:
- What is the expected probability that an agent, a_i, will become a candidate, where $0 \leq i \leq n - 1$?
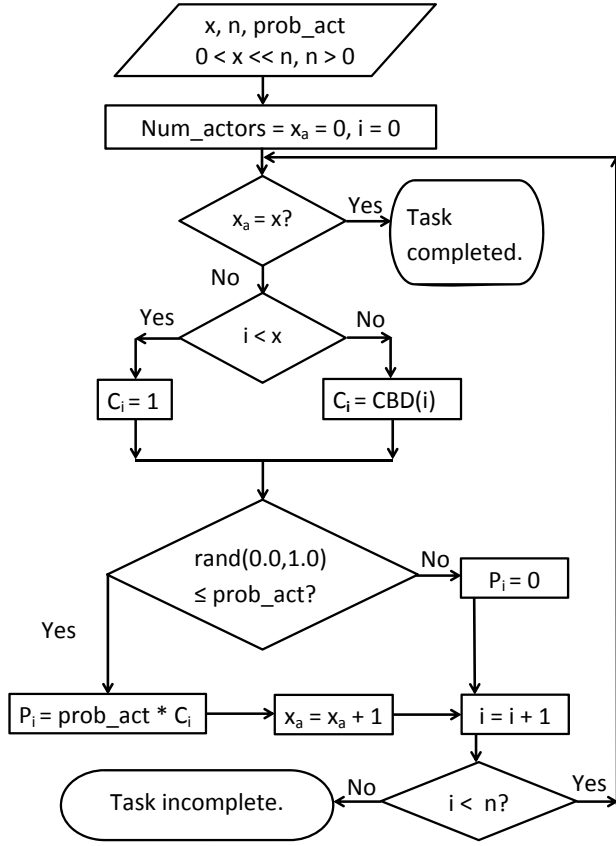
Fig. 2. Flowchart describing the steps that make up a single trial of our system. An agent becomes a candidate if the required x actors have not been obtained. Once a candidate, an agent becomes an actor with probability $prob\_act$. This process is repeated until x agents act or we run out of agents.

- What is the expected probability that an agent, a_i, will become an actor, where $0 \leq i \leq n - 1$?
- Given $prob\_act$ and x, what is the expected number of trials needed for an agent a_i to act at least once?
- Given a fixed number of trials, what value of $prob\_act$ is needed for an agent a_i to gain experience?

The ability to answer these questions will provide guidance as to how one can build response threshold MAS teams that are able to maintain a desired experience pool.

IV. ANALYSIS

Our analysis can be broken down into two cases, the first in which $prob\_act = 1.0$ and the second in which $prob\_act < 1.0$. Because of the presence of threshold-based ordering among agents, we must calculate separate C_i and P_i values for $0 < i < x$ and $x \leq i < n$. Once we calculate P_i, we can use it to calculate the estimated response probability needed to ensure that a given number of agents will gain experience. Table I summarizes the calculations which will be presented in the following sections.

		$0 \leq i < x$		$x \leq i < n$	
		C_i	P_i	C_i	P_i
$prob\_act$	< 1.0	1	$prob\_act$	CBD(i)	$prob\_act$*CBD(i)
	$= 1.0$	1	1	0	0

A. Calculating P_i when $prob\_act$ is equal to 1.0

When $prob\_act = 1.0$, all agents that become candidates become actors. Since x agents are required to complete the task, agents a_0 through a_{x-1} will always become both candidates and actors. Consequently, agents a_x through a_{n-1} do not ever become candidates or actors. Over multiple trials, the total number of agents with experience, x_a, is equal to x agents. The probability that agent a_i gains experience acting on the task, as shown in the second row of Table I, is

$$P_i = \begin{cases} 100\% & \text{if } i < x \\ 0\% & \text{if } i \geq x \end{cases}$$

B. Calculating P_i when $prob\_act$ is less than 1.0

When $prob\_act < 1.0$, not all agents that become candidates will become actors. The following subsections describe and model the processes by which an agent becomes a candidate and then an actor when $prob\_act < 1.0$.

1) Entering the Candidate state: Due to the ordering constraint, the first x agents, a_0 through a_{x-1}, always become candidates as shown in Figure 2. This is shown by the left branch of the decision node $i < x$, where C_i equal to one. For agent $a_i, i \geq x$, the probability that it will become a candidate is given by the Cumulative Binomial Distribution up until i. This is shown by the right branch of the decision node $i < x$ and is indicated in the first row of Table I as $CBD(i)$

$$C_i = \sum_{k=0}^{x} \binom{i}{k} prob\_act^k (1 - prob\_act)^{i-k}, \ i \geq x$$

Figure 3 plots C_i for n equal to 100 and x equal to 45 with $prob\_act$ values varying from 0.1 to 1.0 with a 0.1 increment. The first 45 agents always become candidates and, therefore, their C_i is equal to 1. The C_i for remaining agents is less than 1 as given by the cumulative distribution function.

2) Entering the Actor state: When $prob\_act$ is less than 1.0, agents chosen to become candidates are not guaranteed to become actors. While only x agents will act in any one trial, the same agents may not act from one trial to the next. As a result, over multiple trials we expect the number of agents that gain experience to be greater than x. The expression for P_i, as shown in the first row of Table I,

$$P_i = prob\_act \times C_i$$

Equivalently,

$$P_i = prob\_act \times \left(\sum_{k=0}^{x} \binom{i}{k} prob\_act^k (1 - prob\_act)^{i-k} \right)$$
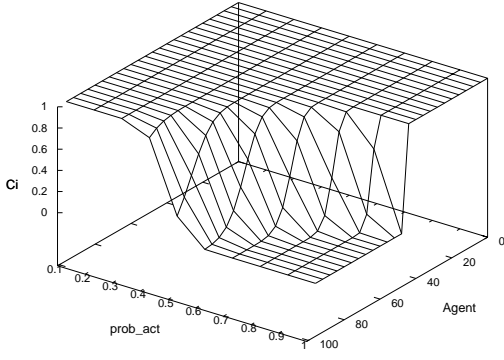
Fig. 3. Expected probability of an agent becoming a candidate estimated using Cumulative Binomial Distribution. The expected probability of agents a_0 to a_{x-1} becoming a candidate is equal to 1; these always become candidates. The expected probability of agents a_x to a_{n-1} becoming a candidate is equal to C_i. Here $x = 45$ and $n = 100$.
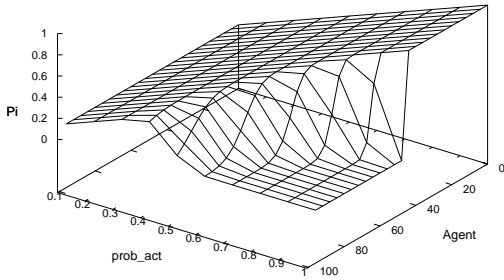

Fig. 4. Expected probability of an agent becoming an actor. The expected probability of agents a_0 to a_{x-1} becoming an actor is equal to the $prob\_act$ value. The expected probability of agents a_x to a_{n-1} becoming an actor is equal to $prob\_act * C_i$. Here $x = 45$ and $n = 100$.

Figure 4 plots P_i for $n = 100$ and $x = 45$ with $prob\_act$ values varying from 0.1 to 1.0 with a 0.1 increment. Because $C_i = 1.0$ for the first x agents, their $P_i = prob\_act$. The remaining agents become actors with the expected probability equal to the product of $prob\_act$ and their C_i.

Figure 5 shows the mean of total number of agents with experience from an empirical study of $prob\_act$ on an agent based simulation. The simulation consists of $n = 100$ agents and $x = 45$ required agents. The $prob\_act$ values vary from 0.001 to 1.000 in steps of 0.001. The simulation is observed over 1000 trials. As $prob\_act$ increases we see an increase in the number of agents that gain experience. Around $prob\_act = 0.25 - 0.3$ to a little under $prob\_act = 0.6$, all the agents in the system gain experience. From then on, the total number of agents with experience steadily decreases as agents that become candidates are more likely to become actors. Finally, when $prob\_act$ close or equal to 1.0, exactly 45 agents gain experience.
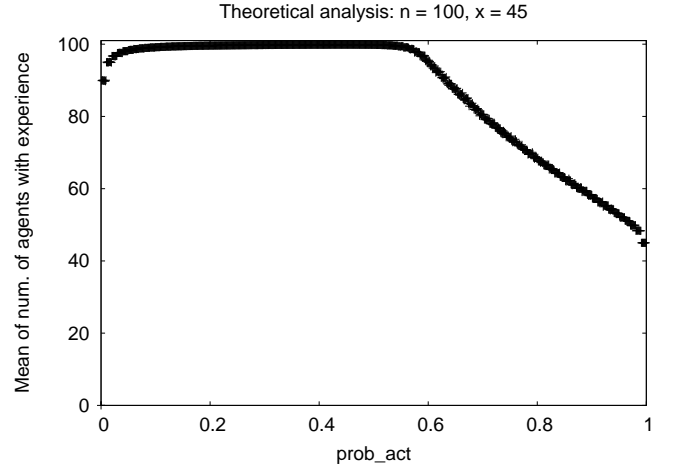

Fig. 5. Mean of total number of agents with experience after 1000 trials for each response probability value. More number of agents gain experience when the $prob\_act$ value is low than when it is high.

C. *Estimating response probability value given P_i*

Given P_i, we calculate the expected number of trials or runs, M_i, necessary for a given agent to act at least once as

$$M_i = \frac{1}{P_i} \qquad (1)$$

Using Equation (1) we can estimate the response probability value for agent a_i to gain experience in a given number of runs. The calculation presented below estimates the $prob\_act$ value for this to occur. In this analysis, s is used in place of $prob\_act$ for clarity. This estimated s value is given by

$$s = \frac{(P_i + iP_i) \pm \sqrt{((P_i + iP_i)^2 - 4(i)(P_i))}}{2 * i} \qquad (2)$$

Equation (2) is obtained by considering the given M_i value to be the least number of runs in which agent a_i acts.

$$M_i = \frac{1}{s \times \left(\sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \right)}$$

The equation of M_i is then differentiated to obtain its slope.

$$
\begin{aligned}
\frac{dM_i}{ds} &= \frac{d\left(\dfrac{1}{s \times \left(\sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \right)} \right)}{ds} \\
&= \frac{d\left(\dfrac{1}{P_i} \right)}{ds} \\
&= -\frac{1}{P_i^2} * \frac{dP_i}{ds} \\
&= -\frac{1}{P_i^2} \sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \left(\frac{(1+k) - s(1+i)}{1-s} \right)
\end{aligned}
$$

The resulting equation when set to zero denotes the minimum point on the M_i curve.

$$\sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \left(\frac{(1+k) - s(1+i)}{1-s} \right) = 0 \quad (3)$$

Solving (3), we obtain the equation that denotes the condition for agent i to gain experience in least number of runs.

$$\sum_{k=0}^{x-1} \binom{i}{k} ks^k (1-s)^{i-k} = (s(1+i) - 1) \sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \tag{4}$$

In equation (4), $\sum_{k=0}^{x-1} \binom{i}{k} ks^k (1-s)^{i-k}$ denotes the mean of values up until i.

This can be approximated as $i*s$ using the approximation to the Cumulative Binomial Density function (CDF). This results in

$$i*s = (s(1+i) - 1) \sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k}$$

$$\sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} = \frac{(is)}{(s(1+i) - 1)} \tag{5}$$

Next, using equation (5) in the equation of P_i, the following quadratic equation in s is obtained.

$$
\begin{aligned}
P_i &= s * \sum_{k=0}^{x-1} \binom{i}{k} s^k (1-s)^{i-k} \\
&= s * \frac{(is)}{(s(1+i) - 1)} \\
P_i(s(1+i) - 1) &= is^2 \\
is^2 - s(P_i + iP_i) + P_i &= 0
\end{aligned}
$$

Thus, the value of s is estimated by the roots of this quadratic equation s_1 and s_2, where

$$s_1 = \frac{(P_i + iP_i) - \sqrt{((P_i + iP_i)^2 - 4(i)(P_i))}}{2*i} \tag{6}$$

$$s_2 = \frac{(P_i + iP_i) + \sqrt{((P_i + iP_i)^2 - 4(i)(P_i))}}{2*i} \tag{7}$$

Due to the approximation using the CDF, it is necessary that the following constraint be placed

$$P_i * i \geq 5$$

The prerequisites for obtaining an estimated s value are that the number of trials in which the agents are expected to gain experience be known and the approximation using CDF constraint be met.

Figures 6 and 7 show the estimated s_1 and s_2 values for agents i=45 to i=99. In both plots, the x-axis shows the agents and y-axis measures the estimated s_1 and s_2 values. We next use these estimated values in an empirical study to determine agents' gain of experience consisting 100 simulations of 11 trials each in which n=100, x=45 and P_i=0.091. We expect
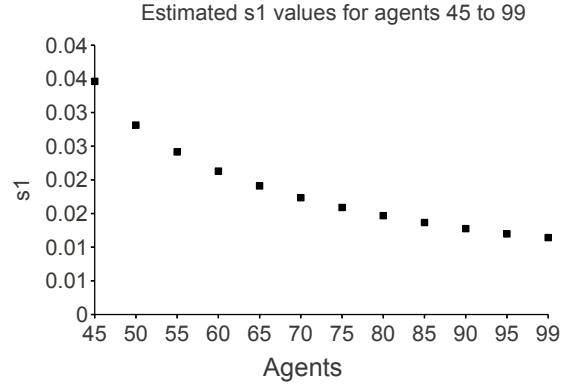

Fig. 6. Estimated s_1 values for agents i, $x \leq i \leq n-1$ with $P_i = 0.091$. Here $x = 45$ and $n = 100$.
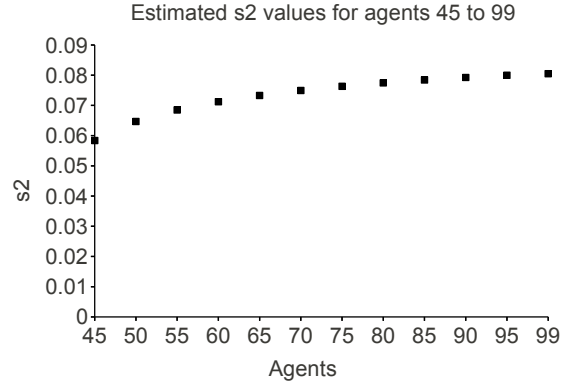

Fig. 7. Estimated s_2 values for agents i, $x \leq i \leq n-1$ with $P_i = 0.091$. Here $x = 45$ and $n = 100$.

agents to gain experience at least once and expect the curve in a plot showing this to be close to one.

Figure 8 plots the average number of times an agent acts in 11 trials when using the corresponding s_1 value from Figure 6. Each black square is averaged over 100 simulations and the error bars show the standard deviation. The x-axis indicates the agent number and the y-axis indicates the average number of times acted. As the agent number increases from 45 to 99, the
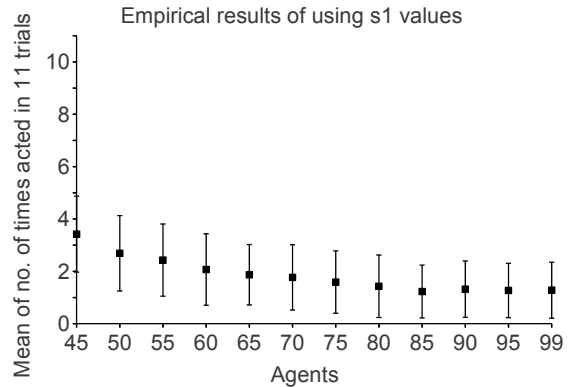

Fig. 8. A plot of the empirical results of using s_1 values from Figure 6 showing the mean of the number of times agent i acts in 11 trials over 100 simulations.
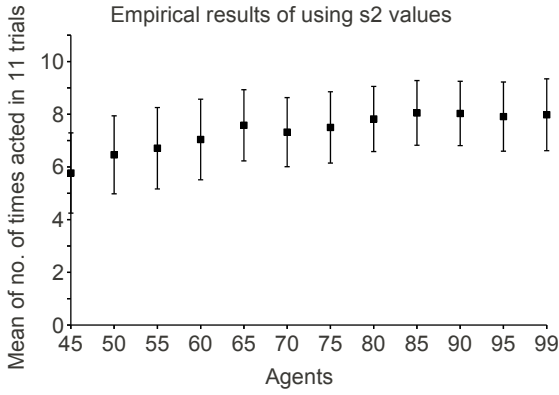
Fig. 9. A plot of the empirical results of using s_2 values from Figure 7 showing the mean of the number of times agent i acts in 11 trials over 100 simulations.

average number of times each agent acts decreases gradually from just under 3.5 to approximately 1.5. As expected, all agents on average act at least once and all values are close to one. These empirical results indicate that s_1 is a good estimation of the probability needed for a given agent to act at least once within a specified number of trials.

Figure 9 plots the same data as Figure 8 but for s_2. Interestingly, as the agent number increases from 45 to 99, the average number of times each agent acts increases from approximately six to just over seven times. The data generated from using s_2 instead of s_1 shows an opposite trend as agent number increases and results in a much higher count in the number of times an agent acts. We do not have a clear explanation for these differences as yet. We speculate that they may be due to the conficting forces between the probability that an agent will become a candidate and the probability that a candidate will act, and are investigating this as well other explanations for this anomaly.

In summary, this section presents the mathematical analysis along with supportive empirical results of variables of interest that could possibly aid in obtaining more than the required number of agents for a given task. We describe equations for the expected probability of an agent becoming a candidate, the expected probability of an agent becoming an actor, and an estimation of the $prob\_act$ value when the number of runs in which agents are required to gain experience is known.

V. EXAMPLE SCENARIO

Imagine an 18th century wagon train consisting of travellers. The travellers must set up camp every night when the group halts. Each night only a limited number of travellers need to participate in the set up. Each traveller has a willingness level for set up. The most willing travellers are usually the first to set up and the least willing may or may not get an opportunity to particiapte in the set up. Each time a traveller participates in the set up, he gains experience that allows him to work faster the next time. If the most willing travellers are always the ones that participate, then over a period of time, those will be the only individuals with experience and the

wagon train is completely dependent on them. If, however, those most willing individuals do not always participate in the set up, then other individuals may have an opportunity to participate, gain experience and the wagon train will have a larger pool of travellers with experience in the set up.

We next present an empirical analysis of the performance of two MAS based on this wagon train scenario with insertion and removal of travellers. Performance is measured by the average of the work speed of all travellers at each trial. If one MAS has a lower average work speed than the other, then its performance as a group, or *group performance*, is better.

Our two MASs are Set A and Set B. Here, we consider a traveller as an agent. In Set A, response probability is 1.0 and the most willing travellers always participate in the set up each night. In Set B, response probability is less than 1.0 and the most willing travellers participate with some probability in the set up each night. Our experiment consists of 100 agents in each set. The set up activity requires 14 agents. The work speed of all agents is initialized to 1.0. Each time an agent acts, its work speed is multiplied by 0.9. Participation in the set up for one night constitutes one trial. If a traveller participates in the set up for more than one night, then this constitutes participation in multiple trials. One simulation consists of 100 trials. We perform 10 such simulations.

The group performance is observed in four cases:

1) *Base case*: We determine the effect of $prob\_act$.
2) *Base case with insertion of agents*: We combine $prob\_act$ with insertion of agents. Let us denote an inserted agent as *new agent* and the agent before which it is inserted as *existing agent*. A new agent has a work speed equal to 1.0. The probability with which the insertion routine in each trial is executed is denoted as *trial insertion probability* and within which the probability to insert a new agent before an existing agent as *agent insertion probability*. Here, trial insertion probability is 0.6 and agent insertion probability is 0.1.
3) *Base case with removal of agents*: We combine $prob\_act$ with removal of agents. The probability with which the removal routine in each trial is executed is denoted as *trial removal probability* and within which the probability to remove an agent as *agent removal probability*. Here, trial removal probability is 0.5 and agent removal probability is 0.1.
4) *Base case with insertion and removal of agents*: We combine $prob\_act$, insertion and removal of agents. The routine to insert an agent is executed every 10 trials and the routine to remove an agent is executed every 15 trials. This mechanism allows us to observe the combined effect of insertion and removal every 30 trials on group performance.

VI. RESULTS

The plots shown in this section have been selected to show the behavior of Set B at three prominent $prob\_act$ values of 0.1, 0.5, and 0.9 and of Set A, though the simulations were run for all $prob\_act$ values from 0.1 to 1.0 in increments of 0.1.

The reason for choosing these values is that at $prob\_act$ equal to 0.1, an agent is most likely not to act. This gives other agents a better chance to act. At $prob\_act$ equal to 0.5, an agent is equally likely to either act or not act on a task. At $prob\_act$ equal to 0.9, an candidate agent is more likely to become an actor. This gives other agents a lesser chance to act. The y-axis measures the average of average group performance over 10 simulations, where each simulation consists of 100 trials. The x-axis denotes the number of trials. The curve labelled *Set A* shows the group performance of Set A. The curve labelled *Set B* shows the group performance of Set B.

Figure 10 shows our base case in which we present the effect of agents responding stochastically at $prob\_act$ values equal to 0.1, 0.5, and 0.9. The group performance of Set A does not vary with $prob\_act$ and reaches a plateau at around 0.86 as shown by curve labelled Set A. This is because only the first 14 agents gain experience on each trial and as a result, the value of their individual work speed approaches zero over multiple trials while the remaining agents' work speed remains unchanged at 1.0. As no further improvement in the group performance of Set A is possible, the average of the average group work speed does not get better than 0.86. Therefore, the base case group performance value for Set A is 0.86.

The group performance of Set B varies with $prob\_act$ as shown by the curve labelled Set B in all three plots. At the end of the simulation, the average of the average group work speed is approximately 0.4, 0.7, and 0.8 for $prob\_act$ equal to 0.1, 0.5, and 0.9 respectively. This is because as more agents gain experience, the group work speed decreases. Therefore, the base case group performance values for Set B at $prob\_act = 0.1$ is 0.4, $prob\_act = 0.5$ is 0.7 and $prob\_act = 0.9$ is 0.8. Since these values are lower than the base case group performance value for Set A, Set B has better performance. As seen from the plots, the difference between the group performances is most prominent when $prob\_act$ is 0.1, moderately prominent at 0.5 and least prominent at 0.9. As expected, this trend shows that at low $prob\_act$ values most agents gain experience, conversely, at high $prob\_act$ values some agents may not gain experience.

Figure 11 shows the second case in which we present the effect of response probability and insertion of agents at every trial. As compared to the base case group performance value of 0.86, the group performance value for Set A is approximately 0.95. As compared to the base case group performance value of 0.4, 0.7, and 0.8 for $prob\_act$ values of 0.1, 0.5, and 0.9 respectively, the group performance values for Set B are approximately 0.5, 0.8, and 0.9 for $prob\_act$ values of 0.1, 0.5, and 0.9 respectively. We observe an increase in the group performance values as compared to the base case. This is because though the number of agents have increased, the number of required agents remain the same and this increases the number of non-participating agents leading to an increase in the average group performance value. In this scenario, Set B is able to perform better than Set A as these new agents may also gain experience and contribute to the overall group performance. This is only possible in Set A if a new agent is
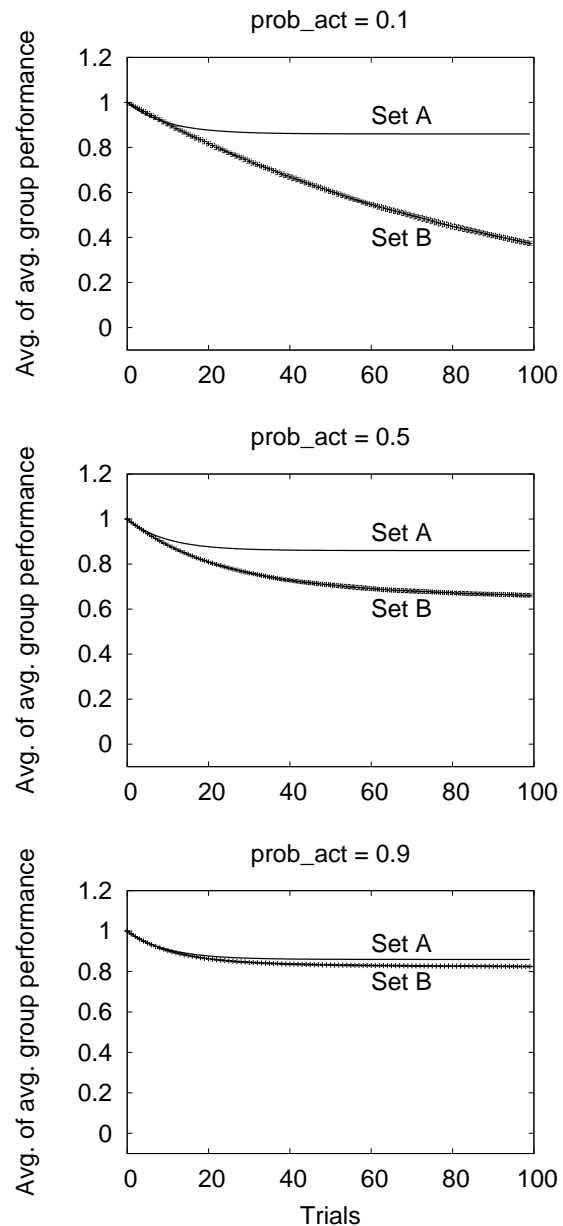

Fig. 10. Base case: effect of response probability on work speed when response probability is equal to 0.1, 0.5, and 0.9 respectively. At low values, Set B has a comparably better work speed than Set A as more agents gain experience. At $prob\_act$=0.9, Set B is slightly better than Set A as less agents gain experience.

inserted before any of the first 14 agents. This, however, has a little effect on group performance as only the first 14 agents participate in the set up.

Figure 12 shows the third case in which we present the effect of response probability and removal of agents at every trial. As compared to the base case group performance value of 0.86, the group performance value for Set A is approximately equal to zero. As compared to the base case group performance value of 0.4, 0.7, and 0.8 for $prob\_act$ values of 0.1, 0.5, and 0.9 respectively, the group performance values for Set B are
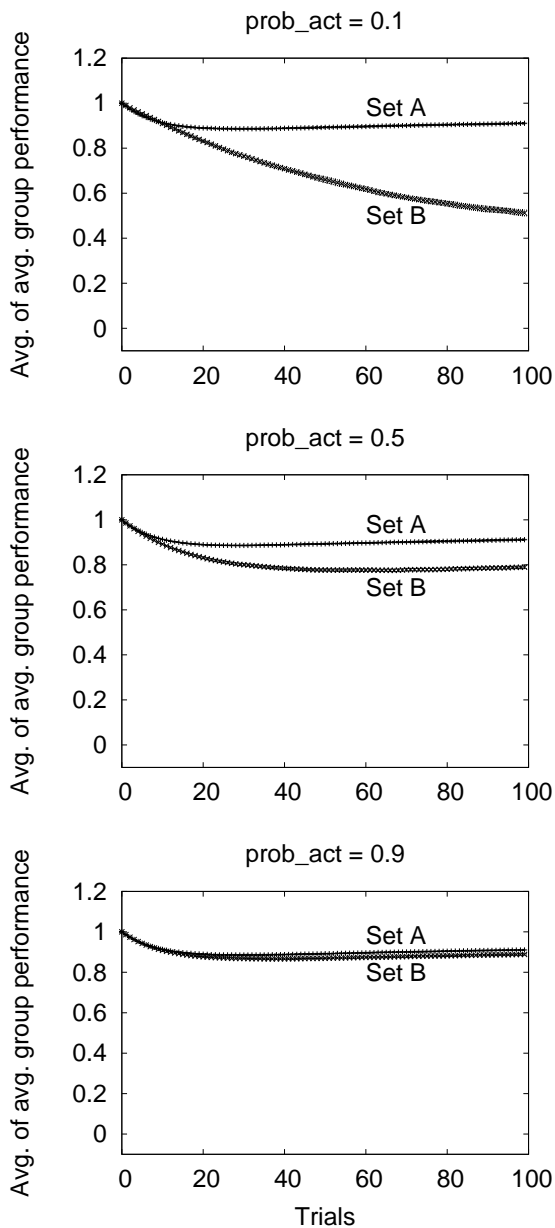
Fig. 11. Effect of inserting agents at every trial on group performance when response probability values are 0.1, 0.5, and 0.9. Insertion of agents at every timestep decreases the overall group performance of both sets, in that the average values with low standard deviations (< 0.001) are higher as compared to the base case. The decrease in performance is a result of increase in non-participating agents. Overall, Set B has lower, and therefore, better group performance than Set A.
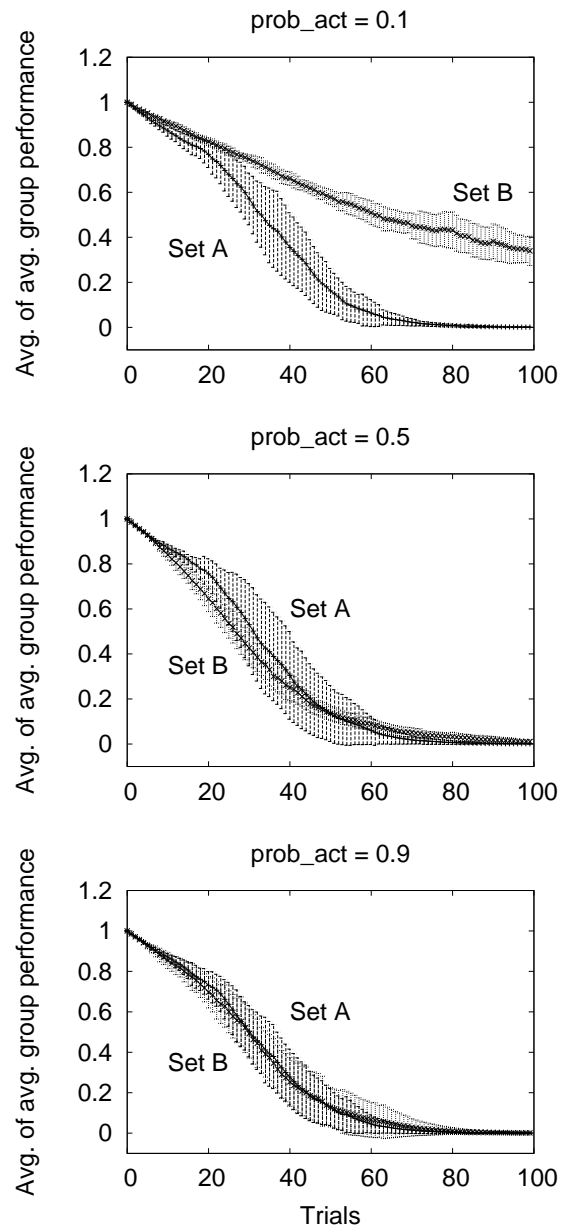
Fig. 12. Effect of removing agents at every trial on group performance when response probability values are 0.1, 0.5, and 0.9. Removal of agents at everytime step increases group performance possibly due to removal of non-performing agents and the average values are lower as compared to the base case. Large standard deviations (< 0.05) indicate that these sets are less stable to agent loss at each trial. For $prob\_act$ of 0.1, Set A performs better than Set B due to a constant presence of participating agents that maintain a lower group work speed. At other instances, these two sets have comparable performances.

approximately 0.3 for $prob\_act = 0.1$ and close to zero at $prob\_act$ equals 0.5 and 0.9. We observe a decrease in the group performance values as compared to the base case. This may be attributed to constant reduction in the group size which has a potential to remove non-performing agents.

At $prob\_act$ equal to 0.1, Set B does not perform better than Set A. We explain this behavior as follows. In Set A, the first 14 agents always act. Now if one among these agents is lost, then the next agent in the ordering acts. This mechanism of including the next in the ordering has the advantage that there exists only one extra agent with a higher work speed than the other 13 actors' work speed. If another actor is removed, then the next in the ordering is chosen. This mechanism allows for a cushioning effect in that majority of the actors have a lower work speed as compared to the newly added actors. This effect, however, is hard to achieve for agents in Set B at a low $prob\_act$ value. This is because the agents chosen to act have a low probablity of acting, which for analytical purposes may be assumed as inaction. Along with this, actors of one trial are not guaranteed to act in another trial. Thus, if an experienced agent is removed, then a new agent chosen to act may or may not have had previous experience. This affects the overall group performance and Set B as a group is less capable of adapting to rapid change as seen in this situation.

A lower group performance in the case of removing agents at each trial does not mean that these sets perform better as one runs the risk of losing most of the agents leading to task incompletion. We present a set of results when agents are removed at every 45th and 98th trial when $prob\_act$ is equal to 0.1. At these longer interval, we expect to see the effect of having a redundant experience pool of agents. As seen from the plots shown in Figure 13, these removals have little effect and the group performances of Set A and Set B are comparable to their base case performance, in that at the end of the simulation these values are close to 0.86 and 0.7 respectively. Also, Set B has better group performance than Set A. Therefore, if an MAS has a redundant experience pool of agents, then it has a potential to promote its robustness.

Figure 14 shows the final case in which we present the effect of agents responding with $prob\_act$ coupled with insertion and removal of agents. Insertion every 10 trials and removal every 15 trials allow us to observe a combined effect every 30 trials. At these instances, the value of the overall group performance slightly increases possibly as a result of loss of experienced and gain of inexperienced agents. The average of the average group performances of both sets are comparable to the base case values, in that at the end of the 10 simulations Set A's group performance is approximately 0.86, Set B's group performance is approximately 0.4, 0.6 and 0.8 for $prob\_act$ values of 0.1, 0.5 and 0.9 respectively. Overall, Set B has lower, therefore better, group performance value than Set A.

In summary, our results support our hypothesis that when agents respond probabilistically, there is an increase in the number of agents that gain experience. The increase is more prominent when response probability is lower and less prominent when response probability is higher. This redundant
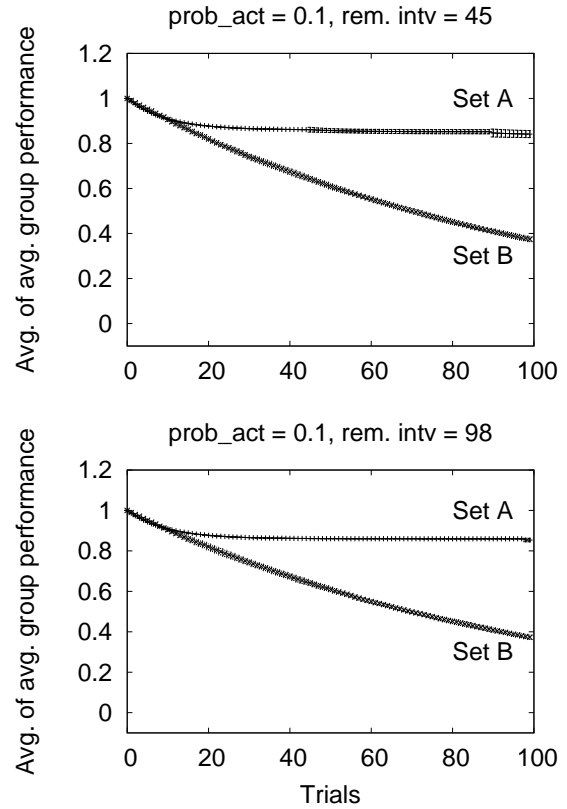

Fig. 13. Effect of removing agents at every 45th and 98th on group performance when response probability value is 0.5. These plots show that when the sets have generated a pool of agents with experience, then removal of agents at larger intervals has little effect. The sets performance is comparable to their base cases and Set B performs better than Set A.

experience pool of agents promotes system robustness under varying conditions of insertion of new and loss of agents. Except under the extreme condition of losing one agent at every time step, a system consisting of agents that respond stochastically consistently performs better than a system that consists of agents that do not respond stochastically.

VII. CONCLUSION

One of the expected benefits of multi-agent systems is their capacity for redundancy which is expected to promote system robustness. We examine the impact probabilistic action has, beyond response threshold, on generating redundancy in multi-agent systems. We present a methodology to introduce redundancy in a multi-agent system consisting of collaborating agents with no direct inter-agent communication. This method explores how probabilistic action in a set of ordered agents affects the system's ability to create a redundant pool of agents with experience.

We present a mathematical analysis of the effect of response probability on the system. The analysis indicates an increase in the number of agents with experience as response probability decreases below 1.0. We provide calculations for the expected probability of an agent becoming a candidate, the expected
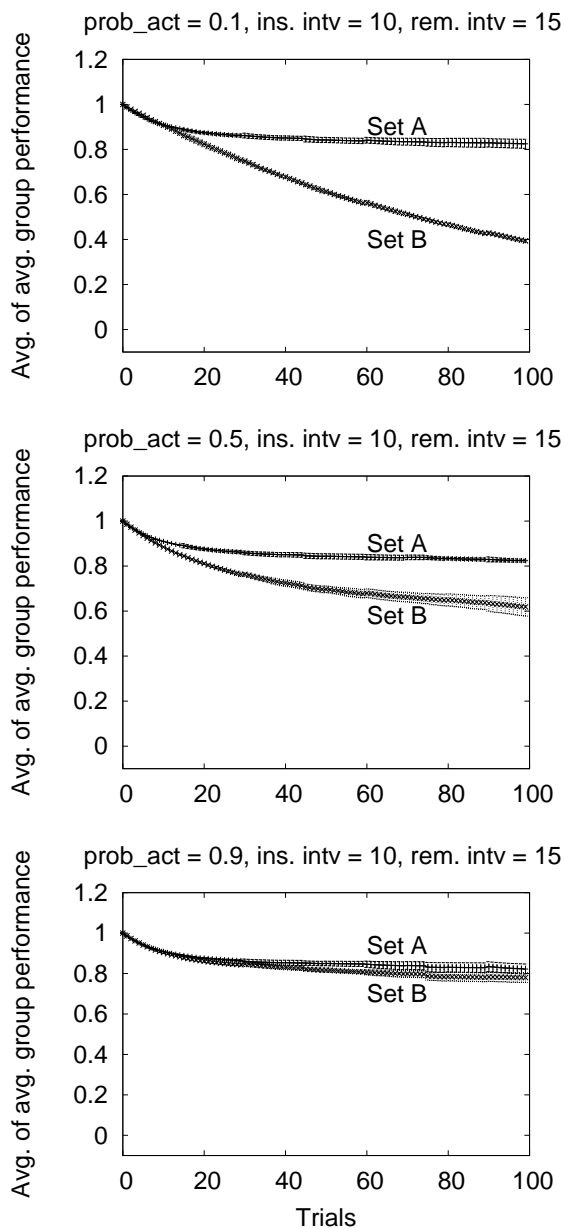
Fig. 14. Effect of combining probabilistic action, insertion and removal of agents on group work speed when response probability is equal to 0.1, 0.5, and 0.9, insertion interval is 10 and removal interval is 15. The results are comparable to the base case results, in that Set B has a lower, and therefore better, group performance than Set A in all three cases.

probability of an agent becoming an actor, the expected number of runs in which an agent gains experience at least once, and finally, an estimation of response probability that can be used when the number of runs in which it is expected to gain experience is known a priori.

Empirical studies investigate the effects of response probability on the size of experience pool in a static scenario as well as a dynamic scenario where there is loss or gain of agents in the group. Results support the hypothesis that a probabilistic response can allow a team of agents to build a pool of agents

with experience. This pool of experienced agents results in a robust system in uncertain environments that may include disturbances such as insertion and removal of agents. The factors that influence the gain of experience by an agent are found to be the agent's position in the ordering, the number of agents required to complete the task, the total number of agents in the system and the response probability value. The estimation of response probability using the equation derived here shows that if actual response probability values are not known, then this estimation may be used to obtain similar results as using the actual values.

In conclusion, response probability has the ability to generate redundancy in a multi-agent system when combined with response threshold. This feature has a promising aspect, in that the variations that are present in the agents as a result of fabrication or calibration may be viewed in a positive way as a feature that produces useful inter-agent variations rather than viewed in a negative manner as errors that must be eliminated.

REFERENCES

[1] S. A. Curtis, M. L. Rilee, P. E. Clark, and G. C. Marr, "Use of swarm intelligence in spacecraft constellations for the resource exploration of the asteroid belt," in *Proc. 3rd International Workshop on Satellite Constellations and Formation Flying*, 2003.

[2] M. J. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," in *Robotics and Autonomous Systems*, vol. 30, no. 1-2, 2000, pp. 65–84.

[3] A. Weidenmuller, "From individual behavior to collective structure: Pollen collection and nest climate control in social bees," Ph.D. dissertation, University of Wurzburg, Germany, 2001.

[4] C. Jones and M. J. Mataric, "Adaptive division of labor in large-scale minimalist multi-robot systems," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, Las Vegas, USA, 2003, pp. 1969–1974.

[5] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Fixed response thresholds and the regulation of division of labor in insect societies," *Bulletin of Mathematical Biology*, vol. 60, pp. 753–807, 1998.

[6] M. Campos, E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Dynamic scheduling and division of labor in social insects," *Adaptive Behavior*, vol. 8, no. 2, pp. 83–95, 2000.

[7] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Comp. Sys.*, vol. 16, pp. 851–871, 2000.

[8] H. A. Reijers, M. H. Jansen-Vullers, M. zur Muehlen, and W. Appl, "Workflow management systems + swarm intelligence = dynamic task assignment for emergency management applications," in *Business Process Management'07*, 2007, pp. 125–140.

[9] X. Yu and B. Ram, "Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups," *Intl. Jrnl. of Production Research*, vol. 44, no. 22, pp. 4793–4813, 2006.

[10] D. Merkle, M. Middendorf, and A. Scheidler, "Self-organized task allocation for service tasks in computing systems with reconfigurable components," *J Math. Model. & Alg.*, vol. 7, no. 2, pp. 237–254, 2008.

[11] W. Xiang and H. P. Lee, "Ant colony intelligence in multi-agent dynamic manufacturing scheduling," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 1, pp. 73–85, 2008.

[12] A. Scheidler, A. Brutschy, K. Diwold, D. Merkle, and M. Middendorf, "Ant inspired methods for organic computing," in *Organic Computing - A paradigm shift for complex systems*, 2011, pp. 95–109.

[13] R. Price and P. Tiňo, "Evaluation of adaptive nature inspired task allocation against alternate decentralised multiagent strategies," in *Proceedings of Parallel Problem Solving from Nature*, 2004, pp. 982–990.