Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

Effects of Task Consideration Order on Decentralized Task Allocation Using Time-Variant Response Thresholds

Annie S. Wu and **Vera A. Kazakova**
University of Central Florida
Orlando, FL 32816-2362
aswu@cs.ucf.edu, kazakova@cs.ucf.edu

Abstract

In this work, we investigate how the order in which tasks are considered to be acted upon by probabilistic agents with time-variant response thresholds within a fully decentralized multiagent system may affect task allocation, task fulfillment levels, specialization tendencies, and system robustness. The tested ordering schemas are: (1) ascending subjective action threshold, (2) descending objective task stimulus, (3) descending action probability P as defined by (Theraulaz, Bonabeau, and Deneubourg 1998) which combines both subjective threshold and objective stimulus, and (4) a random ordering to serve as a baseline for comparison. As the behavior of real-world systems tends to stem from complex interactions of multiple system aspects, we expand our analysis by further breaking down each of these ordering approaches into positive vs. negative reinforcement under different learning and forgetting rates for updating agent response thresholds.

Introduction

Multiagent systems (MAS) are often promoted as better alternatives to complex single agent systems due to their distributed nature and amenability to simpler agents, potentially allowing for increased robustness and decreased costs. However, there are also tasks that simply cannot be accomplished by a single agent, necessitating a cooperative multiagent approach. For example, large scale herding or flanking problems require multiple agents due to the sheer size and number of the elements being herded. Similarly, problems such as perimeter protection, cooperative transport, and cooperative collection are examples of domains that may require the concurrent activity of multiple agents because of spatial demands or limitations of individual agent's capabilities. In such problems, effective coordination among the agents is crucial to successful task completion, but in scalable solutions direct communication among the agents often must be limited or even non-existent. Consequently, real world problems often necessitate solutions that rely on implicit coordination and carefully studied and predictable system-wide emergent behavior.

In this paper, we examine how task consideration order may affect distributed task allocation in a decentralized multiagent system. Specifically, we are interested in problems in which a decentralized MAS must achieve an effective division of labor for *group tasks* and *team tasks*, as defined by (Anderson and Franks 2003), where multiple agents are needed to act concurrently on one or more tasks. Team tasks are similar to the *coalition formation* problem in the MAS literature (Gerkey and Mataric 2004; Shehory, Sycara, and Jha 1998; Shehory and Kraus 1998). Agents are the resources available to satisfy task demands, each working on a single task per time step, the system goal being to maximize task completion through the appropriate allocation of resources/agents to demands/tasks at any given moment.

Much of the work in this area is inspired by social insect societies, given their elegant and decentralized multiagent solutions to sophisticated large-scale survival problems such as ant colony defense and foraging, bee hive thermo-regulation, etc. Our system is based on the *response threshold model* proposed by (Bonabeau, Theraulaz, and Deneubourg 1998), which has been shown to correlate with experimental observations of division of labor in insect societies. Extending their fixed response threshold model to include dynamic adaptation of agent threshold values results in specialization among agents within the system (Theraulaz, Bonabeau, and Deneubourg 1998). The resulting *response threshold reinforcement model* has been studied extensively including, but not limited to, examination of colony size effects on worker specialization (Gautraist et al. 2002); the impact of task demands, age, and mortality on worker specialization (Merkle and Middendorf 2004); and the evolution of specialization in systems of reproducing agents (Duarte et al. 2012). While many of these models assume globally available stimuli, e.g. all insects in a nest can see how big the food pile is, specialization has also been shown to occur on the basis of local information (Agassounon and Martinoli 2002). Variations of the response threshold reinforcement model have been applied to a variety of task allocation problems (Castello et al. 2013; Cicirello and Smith 2003; Ferreira Jr., Boffo, and Bazzan 2007; Nouyan 2002).

Most applications of the response threshold reinforcement model appear to assume that agents are offered one task at a time and their job is to decide whether or not to take (or bid for) the available task. If multiple agents want a task, the task "offeror" decides who "wins" the task. However, in many real world problems agents may sense multiple system demands at the same time, and if an agent has more than one response threshold lower than the corresponding task stimulus value, then it will need to choose one from among multiple tasks in demand. While some studies do allow for the possibility of agents having to choose from among multiple tasks, the choice is made randomly (Jeanson et al. 2007; Duarte et al. 2012; Goldingay and Mouric 2013). *We contend that the strategy by which agents select a task from among the available choices is likely to affect the ability of the MAS as a whole to meet all task demands.* Additionally, a centralized approach requiring an entity to select a winner from among multiple bids may limit the scalability of the system due to: (1) requiring communication between the central unit and the other agents, (2) the central agent being a likely bottleneck and a source of vulnerability, (3) larger teams would result in numerous bids to be assessed, and (4) potentially complex bid supporting arguments could require argumentation methodologies to be incorporated into the centralized task assignment algorithm. These issues can be circumvented through a decentralized approach in which task choice decisions lie within each individual agent.

In this work, we examine the effect of *task consideration order* on the ability of a decentralized MAS to meet the demands of one or more tasks. Task order refers to the order in which tasks are considered by the agents during probabilistic action selection: given a random value *rVal* in the range [0.0-1.0), and some ordered set of tasks, if *rVal* is higher than the probability of selecting the first task, the following task is considered, and so on until a probability above *rVal* is found and the corresponding task is acted upon. Consequently, the order of task consideration is important because once a task is selected, any tasks yet to be considered are ignored by that agent until some future point (e.g. end of time-step, selected task completion, etc.). As a result, tasks which come later in the ordering for many agents may rarely if ever receive consideration, which can lead to some tasks monopolizing the agents' time while others get systematically neglected. As task demands increase with respect to available system resources, it is likely that the negative effects resulting from such improper resource allocation will be magnified.

The task order can be a function of numerous system specifics and requirements. The agents may focus on the system's overall need for that task's completion or a local need for the task, and the level of the need may even be dictated by the nature of the task itself (e.g. if there is a fire, it might be wise to disregard any other considerations); the agents may be guided by reinforcement learned behaviors leading them to prefer tasks they've performed before; agents may also be guided by a predilection toward becoming a specialist (by focusing on a single task) or a generalist (by ensuring their time is spread among multiple tasks). In (Theraulaz, Bonabeau, and Deneubourg 1998), authors combine a task's stimulus value and the agent's learning/forgetting-based threshold for that task into a single probability, allowing agents to be simultaneously guided by their own experiences as well as by the system's demands.

For our investigation, we employ the widely used probability formula proposed by (Theraulaz, Bonabeau, and Deneubourg 1998). We assess how the order in which available tasks are considered by probabilistic agents affect task allocation, task fulfillment levels, specialization tendencies, and system robustness. Our fully decentralized multiagent system is composed of probabilistic agents, with reinforcement-based response thresholds. The ordering schemas tested here are: (1) ascending agent action threshold for the task, (2) descending system stimulus for the task, (3) descending action probability P as defined in (Theraulaz, Bonabeau, and Deneubourg 1998) that combines both the agent's task threshold and the system's task stimulus level, and (4) a random ordering to serve as a baseline for the comparison. As the behavior of real-world systems tends to stem from complex interactions of multiple system aspects, we expand our analysis by further breaking down each of these ordering approaches into that based on positive (+R) vs. negative reinforcement (-R) techniques for agents' threshold updates. This distinction allows for the investigation of the effects task orderings have in systems where agents are encouraged to or discouraged from repeatedly focusing on the same tasks. Additionally, we incorporate a parameter to vary how likely agents are to consider switching to a new task on each time step, effectively incorporating a probabilistic response duration, which has been previously defined as a parameter of natural probabilistic behavior, alongside response thresholds and probabilistic action (Weidenmüller 2004). Note, however, that unlike in (Weidenmüller 2004), our threshold and action probability are not decoupled, but instead the probability depends on the threshold, as defined in (Theraulaz, Bonabeau, and Deneubourg 1998).

Borrowed System Specifications

The basic structure of our system is the same as that of the original variable response threshold reinforcement systems (Gautraist et al. 2002; Theraulaz, Bonabeau, and Deneubourg 1998).

Our agent-based system consists of n agents a_i, $i \in [0, n-1]$, who jointly attempt to satisfy the needs of m globally available tasks. Each task T_j, $j \in [0, m-1]$, has an associated stimulus value S_j which is perceived by all agents and indicates the current level of need for that task. Higher stimulus values indicate a greater system need for the task to be acted upon. Each agent has m thresholds, one for each task, such that θ_{ij} is agent a_i's threshold for task T_j.

All agents are capable of working on all tasks, at most one task per time step. In each time step, every agent that is not continuing on its current task will select a task on which to work. For any given task T_j, the probability that agent a_i will choose to work on it is $P_{ij} = S_j^2/(S_j^2 + \theta_{ij}^2)$ [1].

During each timestep, task stimulus values S_j are adjusted by the task demand, σ_j, and by agent contributions from the previous time step. Agents engaged in a task T_j can each contribute α to that task's goal, causing the following update to its stimulus value on any one timestep: $S_j = S_j + \sigma_j - \alpha * E_j$, where E_j is the number of agents engaged in task T_j during that timestep.

[1] Note that the probability that agent a_i will choose to work on task T_j increases **as $\boldsymbol{\theta_{ij}}$ decreases** and **as $\boldsymbol{S_j}$ increases**.

Updated Model Specifications

Where our system diverges from previous works (Gautraist et al. 2002; Theraulaz, Bonabeau, and Deneubourg 1998) is in the dynamics of the agent's action selection process at each timestep.

In most previous studies using the a variable response threshold reinforcement method, agents sense or are offered at most one task in each timestep. Given a task, an idle agent selects or bids for the task with probability P_{ij}. If more than one agent wants to work on the task, a "winning" agent is chosen by the system.

In our approach, agents can simultaneously sense the demands for all system tasks, and thus multiple task-specific probabilities may be triggered in any one timestep. As a result, idle agents may need to choose from among multiple tasks which one, if any, to act upon. The model in (Theraulaz, Bonabeau, and Deneubourg 1998) corresponds to +R, since the threshold for an action/task becomes lower as that action/task is performed, thus making it more likely to be chosen again in the future. In this work, we augment the model by also assessing its characteristics under -R.

An agent, a_i, deciding on a new task will compare a randomly selected $rval \in [0.0, 1.0)$ to its own set of task specific P_{ij} values in some predefined order, stopping as soon as $rval < P_{ij}$ and performing the corresponding task T_j. If *rVal* exceeds all P_{ij} values, agent a_i will remain idle for the current time step. While higher probabilities P_{ij} will always be more likely to be chosen, the order in which task probabilities are considered will affect overall task assignment and fulfillment. We test several ordering strategies and analyze how they affect task allocation, task completion, agent conditioning and specialization levels, system robustness to agent replacement, and efficiency of resource utilization.

We implement basic +R and -R strategies to simulate an agent's propensity to prefer some tasks (or types of tasks) over others. Specifically, we investigate how an agents' inclination toward repeated action achieved through +R may lead to agent specialization within the system while in the presence of other system aspects such as the various task order consideration approaches considered in this work. We compare our findings with agent generalization stemming from -R, which increases agents' propensity to diversify their actions.

+R is implemented through decreasing the agent's response threshold for a task when said task is acted upon, while increasing the thresholds for the other available tasks (thus making them less likely to be selected in the future). Each time an agent acts on a task, it increases its propensity to act on that task in the future by a *learning factor*, ξ, and decreases its propensity to act on other tasks in the future by a *forgetting factor*, ψ. This adaptation occurs through adjustment of the agents' thresholds. Each time an agent a_i is engaged in task T_j, the threshold is decremented by the learning factor: $\theta_{ij} = \theta_{ij} - \xi$, and the thresholds for all other tasks are each incremented by the forgetting factor: $\theta_{ik} = \theta_{ik} + \psi$, $\forall k \neq j$. This matches threshold updates in (Theraulaz, Bonabeau, and Deneubourg 1998).

-R is implemented identically, with the increments and decrements reversed, thus acting on a task will increase the threshold for said task, making it less likely to be selected in the future, while decreasing the thresholds for the other tasks and making each of them a more likely future choice.

In addition to response probability and response thresholds, an additional parameter called response duration has been identified in biological systems which refers to how long an agent remains on a given task after taking it up (Weidenmüller 2004). While response duration is considered an agent specific characteristic in biological systems, we implement it as a system-wide parameter representing the probability that any given agent will consider switching tasks. We employ the same probability across all agents in order to more clearly see the effects of specific response durations on the system behavior.

Experimental Setup

Inspired by complex interaction of system characteristics in natural systems, we examine how the discussed probabilistic threshold-based actions shape overall behavior given variation in task consideration order, reinforcement type, learning-to-forgetting ratios, heterogeneous task requirements, dynamic task requirements, frequency of task reconsideration, and resource availability. The tests were conducted for teams of 10 and 100 agents, with no differences observed between the two.

Task Consideration Ordering Schemas are the main aspect varied across our experiments. We expect that the order in which tasks are considered by the agents will affect the system's ability to effectively distribute and utilize the available resources (i.e. the agent workforce). We consider the following task ordering paradigms:

- $O\theta_t$*: Descending Preference*
  tasks ordered from lowest to highest agent preference θ_{ij}.
- OS_t*: Descending Urgency*
  tasks are ordered from highest to lowest task stimulus S_j; this ordering is independent of θ_{ij}.
- OP–val*: Descending Probability Value (Combination of Preference & Urgency)*
  tasks are ordered from highest to lowest probability P_{ij}.
- $ORand$*: Random Order*
  this ordering is random and used as a baseline.

For each of these ordering schemas, we test and analyze the system dynamics resulting from: (1) +R encouraging repeated actions, thus encouraging specialization, and (2) -R discouraging repeated actions, thus encouraging diversification. As a result, we review a total of eight different setups composed of four different task ordering strategies of two reinforcement types each.

Learning-to-Forgetting Ratios tested here are as follows: (1) learning ξ=0.01, forgetting ψ=0.0033 and (2) learning ξ=0.01, forgetting ψ=0.0016. The reasoning behind the ratios these choices represent is as follows. For the first ratio, given a 1% learning speed, we distribute forgetting evenly across all the other tasks not currently being acted on (excluding T0: idling), leading to a 0.33% forgetting per task in our 4-task setup (again excluding T0). This simulates a finite memory where learning something new requires an equal overall forgetting of something else. Note that this 1-to-3 forgetting-to-learning ratio represents faster forgetting than any of the values tested in (Theraulaz, Bonabeau, and Deneubourg 1998), which ranged from 0-to-3 to 0.5-to-3. To further assess the system's capabilities, we lower the forgetting rate to match a 0.5-to-3 ratio in our second set of experiments ($\psi = 0.0016$).

Heterogeneous Task Requirements are employed to test the system's capacity to appropriately allocate the available resources. We test its performance with a 4 task setup (T1,T2,T3,T4), plus an idling task (T0). The initial demand ratios for the four tasks are 8:5:5:2. These ratios represent the consumption of goods within the system that agents need to fully replenish over the course of a "simulation day". To assess the system's capacity to adapt to a changing environment, we conduct a second set of experiments that begin with the task demand ratios of 8:5:5:2, but change to 2:5:5:8 after half of the allotted time (namely 10 simulation days).

Response Duration is implemented through a probability of reconsidering an agent's current task choice. Thus, a probability of 1.0 leads to choosing a new task on every time step, while a probability of 0.25 only causes reconsideration during one in every four time steps. The tested re-assignment probabilities are: 1.0, 0.75, 0.5, and 0.25.

Resource Availability and their utilization are crucial in decentralized systems. A successful division of labor may have to resolve having insufficient overall resources, as well as not over-reacting to demand when provided with plentiful resources (as idling conserves resources). We test system behavior given the following available-to-needed resource (i.e. agents) ratios: 0.8, 1.0, 1.3, and 2.0.

Experimental Results and Discussion

We first assess the ordering schemas using the following setup: $\xi = 0.01$, $\psi = 0.0033$, and a static task requirement ratio of 8:5:5:2. We'll refer to this as our "basic setup". Each setup is tested for all four orderings, and both reinforcement mechanics, leading to eight tested systems . Each such group of eight is tested with the four possible resource availabilities (80%, 100%, 130%, and 200%), as well as with the four possible task-switching probabilities (1.0, 0.75, 0.5, and 0.25). Twenty runs are performed on each setup.

Ordering Schemas As expected, use of +R vs. -R has a significant effect on the resulting system behavior.

Under R+, task ordering appears to have the greatest effect on the likelihood that agents will specialize on one or more tasks. $O\theta_t$ and $OP\text{–}val$ orderings result in significantly more stable task allocation than OS_t and $ORand$ (see Figure 1). These differences are due to $O\theta_t$ and $OP\text{–}val$ being based on the θ_t values. Under R+, performing an action will decrease θ_t for said action, while increasing it for the other actions. As a result, θ_t values tend to converge toward 0.0 and 1.0, respectively, making θ_t and, consequently, the $O\theta_t$ and $OP\text{–}val$ task orderings relatively stable.

Whenever θ_t reaches zero, the probability becomes $P_t = S_t^2/S_t^2 = 1.0$. Consequently, any time a task with $\theta_t = 0.0$ is selected, the action will follow, regardless of stimulus level. In $O\theta_t$ and $OP\text{–}val$ tasks with lowest thresholds are considered first, likely causing them to be selected given the aforementioned $P_t = 1.0$, and increasing their θ_t further, leading them to be selected again. The resulting feedback loop leads to specialization tendencies, decreased task switching, and increased efficiency.

Under -R, no significant differences in stability are observed for any of the orderings. Under -R, performing an action will increase θ_t for said action, while decreasing it for the other actions. As a result, on the following iteration a different action would be more likely chosen, since probability will be higher for lower θ_t values. Based on system need, some tasks will still be selected more often, increasing their θ_t to a maximum of 1.0. The resulting $P_{ij} \in [0.0, 0.5]$ allows tasks that are not first in the ordering to be acted on as well. This results in a decreased importance of the task consideration order.

Learning-to-Forgetting Ratios When forgetting rate is lowered to match 0.5-to-3 ratio (fastest forgetting tested in (Theraulaz, Bonabeau, and Deneubourg 1998)), specialization is also achieved in the other task consideration orderings, namely OS_t and $ORand$, but only when extra agents are available (again, 130% aned 200%).

Dynamic Task Requirements Altering our basic setup, we switch task demands half way through the simulation, from 8:5:5:2 to 2:5:5:8, thus reversing the demands for T1 and T4 (see Figure 1). Note that there is never any demand for T0:idling. As expected, any setup that settles toward some specialization is temporarily disturbed by the new demands, but settles to a new stable distribution soon after. Setups in which agents continuously switch among tasks (those under OSt and ORand, for -R and +R, as well as Ot and OP-val for -R only) begin switching even more often, creating more inefficiencies as the system attempts to keep up with the altered demand. Note that because demand is reversed but not actually increased, the additional task switching is unnecessary and is less efficient than swapping the earlier T1 and T4 patterns. This showcases that (1) meeting system needs does not imply optimal resource utilization and (2) upon adjusting to new demands, final action selection pattern depends on initial pattern when demands first changed.

Response Duration Variation To vary system-wide response duration, we vary the task re-assignment probability. Starting with a baseline of re-considering the current assignment on every time step (i.e. 100% probability to reconsider), the rate of re-consideration is lowered (to 75%, 50%, and finally 25%), increasing the average duration of individual task assignments. While this decreases the computation required from each agent per time step, it can also lead to decreased responsiveness to system changes, such as when other agents switch between tasks through chance thus altering system needs.

Resource Availability Variation While performance is expected to increase with increasing resources and decrease otherwise, we test different resource levels to assess overall system behavior. The observed general trend is that spe-

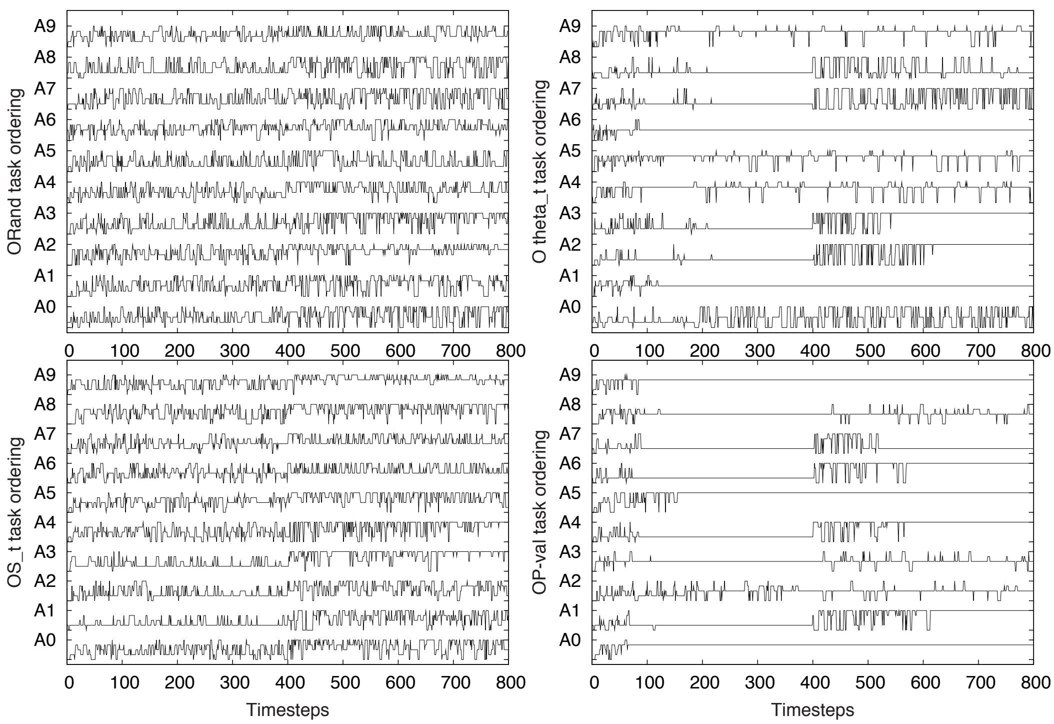

Figure 1: Task on which each agent works in each timestep. The vertical space is subdivided to show the task choices for each of ten agents. The vertical space for each agent is further divided into the five possible tasks, T0-T4. The horizontal axes indicate time steps. The plotted lines show the task choice for each agent in each timestep. Task demands change at step 400.

cialization is most prominent with lowest resources (80% of required agents are available), gradually dissipating as resources increase, and disappearing entirely at the highest tested level (200% of needed agents are available). This trend suggests that the system is capable of leveraging requirement with spreading experience across the population, when efficiency is less crucial.

Given excess resources (130% and 200%), task fulfillment levels are consistently higher for all four orderings using -R than those using +R. This is especially prominent with the highest amount of extra resources/agents. The reason is that under -R, thresholds for all the tasks evolve toward 0.0, except for idling, which instead grows toward 1.0 as a result of all the extra agents choosing to idle. Consequently, when selecting an action, the agents will have a strong preference toward anything but idling, continuously doing work. With +R, thresholds evolve in the opposite direction: $\theta_0 = 0.0$, $\theta_{\forall t \neq 0} = 1.0$, thus leading the agents to idle more, to the point of coming near but not actually fulfilling the task demands despite the additional resources.

Analysis of Probabilistic Response

If an agent has only one task to consider, the original probability formula from (Theraulaz, Bonabeau, and Deneubourg 1998) appears to lead to desirable behavior: the closer an agent's threshold is to zero the higher the probability that the agent will act regardless of stimulus level; as threshold increases, the actual need for the task becomes more relevant. When stimulus and threshold are equal, the probability is exactly half, which is sensible since probability increases when stimulus increases and/or threshold decreases, e.g. $S_j = 0.25$ and $\theta_{ij} = 0.25$ correspond to a 25% task need and a threshold that translates to 75% of an agent's maximum action propensity (maximum propensity is $\theta_{ij} = 0\%$), averaging out to a 50% probable response. This model, however, can result in undesirable behavior in domains where multiple tasks are being considered simultaneously.

When stimulus and threshold are inverted, the underlying task need and propensity match, e.g. $S_j = 0.25$ and $\theta_{ij} = 0.75$ correspond to a 25% need and a 25% propensity. Assuming equal weighting of stimulus and threshold values, the discussed model appears to emphasize action propensity and undervalue system need (25% need and 25% propensity lead to 10% probable response, instead of a more sensible 25% probable response). This valuation leads the system away from utilizing its resources efficiently by not placing enough value on current system need and over-emphasizing propensity to act (threshold). See Table 1 for examples.

Since considering any task with a low threshold will lead to a high probability of acting, an agent may end up neglecting the actual system needs expressed through the stimulus values. An extreme case of this is when $\theta_{ij} = 0$, which results in action probability of 1.0 regardless of stimulus level. As a result, the order of task consideration becomes crucial to the task-selection behavior: starting with highest threshold will lead to nothing else being considered, but starting with a task stimulus could lead to a higher responsiveness to system needs.

Task need	%Max propensity	S_j	θ_{ij}	P-value (P_{ij})
0%	0%	0.0	1	0.0
25%	0%	0.25	1	0.05882
50%	0%	0.5	1	0.2
75%	0%	0.75	1	0.36
100%	0%	1	1	0.5
75%	25%	0.75	0.75	0.5
50%	50%	0.5	0.5	0.5
25%	75%	0.25	0.25	0.5
10%	90%	0.1	0.1	0.5
0%	100%	0.0	0.0	0.5
25%	100%	0.25	0.0	1
50%	100%	0.5	0.0	1
75%	100%	0.75	0.0	1
100%	100%	1	0.0	1
99%	99%	0.99	0.01	0.9999
90%	90%	0.9	0.1	0.9878
75%	75%	0.75	0.25	0.9
25%	25%	0.25	0.75	0.1
10%	10%	0.1	0.9	0.0122
1%	1%	0.01	0.99	0.0001

Table 1: Sample [stimulus, threshold] ([S_j,θ_{ij}]) value pairings and the resulting probability ($P\!-\!value$), calculated as $P_{ij} = S_j^2/(S_j^2 + \theta_{ij}^2)$.

Consequently, we suggest that a new model may be needed for systems where multiple tasks are available at any given time, thus requiring (1) some choice of task consideration order and (2) a probability function that is not liable to ignore the current system need for completing said task in the face of low task threshold values.

Conclusions

In this work, we examine how the order in which tasks are considered for selection in a distributed task allocation problem can affect the resulting task allocation and overall behavior of a decentralized MAS. We hypothesize that the magnitude of effects will be more evident when task demand is high with respect to available resources and show that to be the case, as highest specialization is found in systems provided with only 80% of the required agents. Experimental results are obtained using an agent-based simulation in which agent act independently, do not communicate, and select tasks using the response threshold reinforcement model employing positive or negative reinforcement techniques. Results indicate that when resources/agents are plentiful, task ordering strategy has little effect on the performance of the MAS. Learning to forgetting ratio appears to have a greater effect, allowing for specialization even under less static ordering approaches, such as random.

Positive reinforcement approaches drive probability of acting toward 1.0 for any commonly selected tasks (i.e. those with high θ_{ij}), leading to specialization in task consideration orderings relying on threshold values. Negative reinforcement keeps probability between 0.0 and 0.5, increasing the likelihood of items further down the task ordering list to be selected, promoting generalization.

Interestingly, the simple response threshold reinforcement decision process used in our system appears to produce a very efficient decentralized task allocation behavior. The percent effort devoted to each task over an entire run is approximately proportional to the task demands in all experiments and an appropriate proportion of agents remain idle when available resources exceed task demand.

References

Agassounon, W., and Martinoli, A. 2002. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proc. Int'l Joint Conf. Auto. Agents & Multi-Agent Sys.*

Anderson, C., and Franks, N. R. 2003. Teamwork in animals, robots, & humans. *Advances in the Study of Behavior* 33:1–48.

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

Castello, E.; Yamamoto, T.; Nakamura, Y.; and Ishiguro, H. 2013. Task allocation for a robotic swarm based on an adaptive response threshold model. In *Proc. 13th Int'l Conf. Control, Auto., & Sys.*

Cicirello, V. A., and Smith, S. F. 2003. Distributed coordination of resources via wasp-like agents. In *Proc. 1st Int'l Workshop on Radical Agent Concepts, LNCS 2564*, 71–80.

Duarte, A.; Pen, I.; Keller, L.; and Weissing, F. J. 2012. Evolution of self-organized division of labor in a response threshold model. *Behavioral Ecology and Sociobiology* 66:947–957.

Ferreira Jr., P. R.; Boffo, F. S.; and Bazzan, A. L. C. 2007. Using Swarm-GAP for disstributed task allocation in complex scenarios. In *Proc. Int'l Conf. Autonomous Agents & Multiagent Systems*.

Gautraist, J.; Theraulaz, G.; Deneubourg, J.; and Anderson, C. 2002. Emergent polyethism as a consequence of increased colony size in insect societies. *J. Theor. Biology* 215:363–373.

Gerkey, B. P., and Mataric, M. J. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research* 23(9):939–954.

Goldingay, H., and Mouric, J. V. 2013. The effect of load on agent-based algorithms for distributed task allocation. *Information Sciences* 222:66–80.

Jeanson, R.; Fewell, J. H.; Gorelick, R.; and Bertram, S. M. 2007. Emergence of increased division of labor as a function of group size. *Behav. Ecology & Sociobiology* 62(2):289–298.

Merkle, D., and Middendorf, M. 2004. Dynamic polyethism and competition for tasks in threshold reinforcement models of social insects. *Adaptive Behavior* 12(3):251–262.

Nouyan, S. 2002. Agent-based approach to dynamic task allocation. In *Proc. 3rd Int'l Workshop on Ant Algorithms*, 28–39.

Shehory, O., and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artif. Intel.* 101(1-2):165–200.

Shehory, O. M.; Sycara, K.; and Jha, S. 1998. Multi-agent coordination through coalition formation. In *Intelligent Agents IV, LNCS 1365*, 143–154.

Theraulaz, G.; Bonabeau, E.; and Deneubourg, J. 1998. Response threshold reinforcement and division of labour in insect societies. *Proceedings of the Royal Society B* 265:327–332.

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.