Decentralized Swarm Desynchronization via Inter-agent Variation for Logistical Resupply

Joseph P. Giordano[1], **Annie S. Wu**[1], **Arjun Pherwani**[1], **H. David Mathias**[2]
[1]University of Central Florida, Orlando, FL 32816-2362
[2]University of Wisconsin – La Crosse, La Crosse, WI 54601
aswu@cs.ucf.edu, dmathias@uwlax.edu, {joseph.giordano,arjunp}@knights.ucf.edu

Abstract

Decentralized computational swarms have been used to simulate the workings of insect colonies or hives, often utilizing a response threshold model which underlies agent interaction with dynamic environmental stimuli. Here, we propose a logistics resupply problem in which agents must select from multiple incoming scheduled tasks that generate competing resource demands for workers. This work diverges from previous attempts toward analyzing swarm behaviors by examining relative amounts of stress placed on a multi-agent system in conjunction with two mechanisms of response: variable threshold distribution, or duration level. Further, we demonstrate changes to the general swarm performance's dependence on paired desynchronization type and schedule design, as the result of varied swarm conditions.

Introduction

In this work, we investigate the impact of agent desynchronization on task allocation in decentralized swarms. We focus on swarms consisting of simple threshold-based stimulus-response agents and examine how they respond to dynamically changing task demands in a logistics re-supply problem. Task allocation in a decentralized swarm is a challenging problem because the lack of explicit coordination among agents can make it difficult for agents to distribute themselves appropriately in response to multiple task stimuli. This problem is less apparent in situations where agents only sense local stimuli, but in situations where agents sense one or more global stimuli, there is a risk that all agents will act identically which is inefficient and nullifies many of the benefits of swarms. Desynchronizing the actions of the agents in a swarm can be an effective way of mitigating this problem. Forcing agents to make decisions at different times means that the stimuli they sense may be different which can potentially result in more diverse responses. When synchronized swarms are subjected to identical stimuli, suboptimal behaviors may result from actions for which agents respond in concert. Desynchronization effectively solves this problem by diversifying agents' ability to sense and respond to stimulus.

Biological studies hypothesize that there are multiple ways to desynchronize a swarm of decentralized agents (Jones et al. 2004; Weidenmüller 2004; Weidenmüller, Chen, and Meyer 2019). This work examines two of the approaches identified as relevant in biological swarms, variation in response threshold and variation in response duration, and investigates their impact on the coordination of artificial swarms. *Response threshold* refers to the threshold at which an agent will respond to a given task stimulus. Diversifying the threshold values assigned to agents for a given task causes different agents to consider acting on that task at different times. Low threshold agents will respond when the task stimulus is low; high threshold agents will not respond until the task stimulus is higher. There are many examples in which variation in agent response thresholds successfully helps to coordinate decentralized agents, including studies on both static (dos Santos and Bazzan 2009; Kanakia, Touri, and Correll 2016; Krieger and Billeter 2000; Meyer et al. 2015; Price and Tino 2004; Wu et al. 2020) and dynamic (Campos et al. 2000; Castello et al. 2018; 2013; de Lope, Maravall, and Quinonez 2015; Engholdt, Mathias, and Wu 2020; Goldingay and van Mourik 2013; Price and Tino 2004; Theraulaz, Bonabeau, and Deneubourg 1998) thresholds, as well as agents that behave deterministically (Krieger and Billeter 2000; Wu et al. 2020; Wu and Mathias 2020) and probabilistically (Castello et al. 2013; 2018; Correll 2008; dos Santos and Bazzan 2009; Goldingay and van Mourik 2013; Kazakova and Wu 2018; Niccolini, Innocenti, and Pollini 2010; Nouyan et al. 2005; Pang et al. 2017; Price and Tino 2004; Yang, Chen, and Li 2010). Examination of whether the distribution of the threshold values affect swarm performance finds that simple uniformly distributed thresholds are effective and efficient on dynamic problems where task demands are not known in advance (Wu et al. 2020; Wu and Mathias 2020). *Response duration* refers to the amount of time that an agent works on a task before stopping to check task stimuli and re-evaluate its actions. Not re-evaluating actions at the same time means that different agents may encounter different task stimuli, producing diverse agent responses. Previous systems have used external factors such as the amount of time active or resting (Agassounon and Martinoli 2002; Liu et al. 2007), success rate of agent (Yasuda, Kage, and Ohkura 2014), and perceived task demands (Jones and Mataric 2003) to regulate the duration and timing of agent actions. The response duration we study here, however, is an intrinsic characteristic of an agent, and not an external force. Initial studies on the effects of response duration suggest that it can effectively desynchronize agents in a swarm but may reduce the overall responsiveness of a swarm (Engholdt, Mathias, and Wu 2020).

This work investigates the effect that variation in response threshold and response duration can have on promoting self-organization in decentralized swarms. We test swarm performance on a logistics re-supply problem which consists of a schedule of multiple task demands that vary over time. Each task represents a resource that must be retrieved and the task demands represent the amount needed within a given timestep. Based on sensed task demands, agents in a swarm independently decide what task to take on at any given time. This problem is defined by global environmental stimuli to which the agents in a swarm respond, subject to their threshold and duration of response. Ideally, a swarm will be able to divide the labor of its workers appropriately such that task demands are satisfied in a timely manner. Agents should be used sparingly both to conserve agent resources and to minimize waste resulting from overdelivery. Because task switching may incur physical or time costs, stable distributions of agents with minimal task switching are desirable. Optimal swarm behavior, in this testbed, is provided by agents that perfectly satisfy the requested schedule of material demands. This occurs when agents respond to every need immediately, without excessive task switching or overdelivery. We predict that agent performance will be more consistent when agent desynchronization is highest and that the need for sufficient desynchronization is more critical for more difficult problems. To provide grounds for this investigation, we look at the effect of different types of logistic schedules on agent performance. Problem schedules consisting of multiple sessions with equivalent total task demands with varied session lengths allows for investigation into relations between task frequency and agent behaviors. We examine the interaction of these schedule types with the parameters of response threshold and response duration using two-way analysis of variance (ANOVA).

Implementation

Problem: Our testbed problem is based on the logistics of material resupply. While the *logistics resupply* problem is very general, we consider a specific scenario: a large construction site in which the primary construction activities occur centrally and required materials are stored on the periphery. Agents move materials from their storage locations to the construction zone according to a schedule. The schedule is an input to our simulation.

Each material $m_i,\ i \in \{1, \ldots, M\}$ defines a task d_i that requires moving 1 unit of m_i from the storage location. Every agent is capable of performing all M tasks. A simulation consists of a predetermined number of discrete timesteps, denoted `Max_steps`. Stimulus $\sigma_i(t)$ represents demand for material m_i at timestep t. All stimuli are global and, therefore, known to all agents. During every timestep, each agent selects a task based on the current stimuli. Stimulus demand values depend on a given problem type, and they are subject to fluctuation via agent delivery, as we will see for this system. These task demands are specified in two schedules. The original schedule, S_O, is an input to the simulation. It consists of static demand quantities for each material in each timestep. The dynamic working schedule, S_W, represents updated task demands based on quantities of materials actually delivered during a simulation. Schedules consist of *sessions*; contiguous sequences of fixed non-zero demand for a given material. In S_O, sessions have defined intervals, whereas S_W contains dynamic start and end times to reflect deliveries. Session end times are triggered when the total requested material amount has been delivered. Subsequently, start times may be delayed, since sessions for the same material must remain disjoint in time. Additionally, start times cannot be decreased. To define demand more precisely, let $D_i(t)$ be the demand for m_i at timestep t in S_O. Further, let $A_i(t)$ denote the quantity of m_i delivered by the swarm in timestep t. Then we define the actual demand for m_i in timestep t as $\sigma_i(t) = D_i(t) + \sigma_i(t-1) - A_i(t-1)$ where $\sigma_i(t-1) - A_i(t-1)$ is the delivery deficit from the previous timestep. Altogether, we include 3 benchmarks for measuring swarm performance: Timesteps to completion, sum of agent over-delivery, and average number of task switches per agent - all for a given schedule. The domain goals are to minimize each respective benchmark.

Figure 1: Dynamic task demands for an original schedule (left) and a working schedule (right). Demands are negative in the case of surplus delivery in the preceeding step(s).

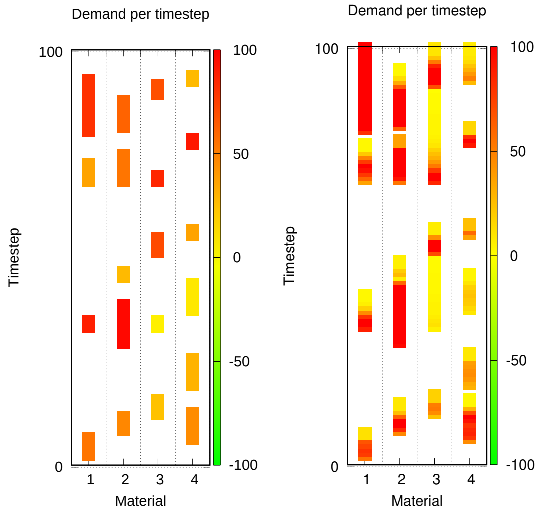

Figure 1 shows heatmaps illustrating task demand quantities over time for an original schedule and a working schedule. Materials appear on the x-axes with timesteps on the left y-axes and demand intensity on the right y-axes. In the original schedule, demand is constant throughout each session. In the working schedule, demand changes to reflect under-delivery or over-delivery in previous timesteps. Demand values within a timestep may be negative, representing a running surplus, as the required material from the previous timestep may have been exceeded.

System Description: In this section, we describe aspects of our implementation that are independent of the testbed

Algorithm 1 Task Selection(Agent i)

```
if random(0, 1) < prob_check then
    $T \leftarrow Array[M]$ initialized to 0
    for $j \leftarrow 1\ to\ M$ do
        if $\sigma_j > \tau_j^i$ then
            $T[j] \leftarrow \sigma_j$
    if $\exists$j s.t. $d_j \in T$ then
        current_task$_i$ $\leftarrow$ random $d_j$
    else
        current_task$_i$ $\leftarrow$ idle
else
    current_task$_i$ $\leftarrow$ prev_task$_i$
```

problem described above. These include forms of inter-agent variation and methods used for agent task selection. The importance of inter-agent variation for swarm goal achievement has been understood for some time (Weidenmüller 2004). In this work, we explore two forms of inter-agent variation: response thresholds and response duration. *Response threshold* defines a minimum value required for an environmental stimulus to trigger a response to that stimulus by an agent. In this work, a response threshold τ_i for task d_i is a value in $[0..1]$. System parameter Scaling_factor allows response threshold values to be applied to problems with significantly different demands or in different domains. Recall that $\sigma_i(t)$ represents the stimulus (demand) for material m_i at time t. Then, an agent may activate for task d_i in timestep t if $\sigma_i(t) \geq \tau_i \cdot \texttt{Scaling\_factor}$. System parameter Thresh_init defines the method by which threshold values are determined. If Thresh_init $\in (0..1]$ then $\forall\ i \in [1, \ldots, M]\ \tau_i = \texttt{Thresh\_init}$. In other words, the threshold values are a single constant for all tasks. Thresh_init > 1, dictates heterogeneous response thresholds generated according to one of several probability distributions. This provides a form of inter-agent variation in which an agent's thresholds for the M tasks may differ and the thresholds for each task may differ across agents. We define τ_i^j to be the response threshold for task d_i for agent j. The Thresh_init value determines the probability distribution used. In this work, the distributions we explore are uniform, Gaussian, and Poisson. prob_check values indicate higher time on task. Agents reevaluate task demands at different times, allowing for variation in their responses due to the possibly different stimuli at those times. Thus, response duration is an effective strategy for desynchronizing swarms in our model. *Task selection* is the process by which agents evaluate current task demands and choose to perform a task to address those demands, or in some cases, remain idle. As described above, an agent performs task selection only if prob_check is satisfied. Candidate tasks are those for which demand exceeds the agent's response threshold. An agent selects one of the candidates uniformly at random. See Algorithm 1 for a more detailed description of task selection. Recall that M represents the number of tasks.

Expt	Popsize	Scaling_factor	Stress_index
A	100	100	1.00
B	100	50	0.50
C	100	25	0.25
D	50	100	2.00
E	150	100	0.67
F	200	100	0.50
G	400	100	0.25

Table 1: Seven experiments, varying stress index values along two axes: Fixed population with decreasing scaling factor and fixed scaling factor with increasing population.

Results

The experimental study is divided into response threshold and response duration segments. For response thresholds, we examine 7 probability distributions for generating values: constant, uniform, Gaussian ($\mu = 0.50, 0.25$), and Poisson ($\lambda = 3, 5, 7$). For response duration, we test homogeneous prob_check values from [0.1,1.0], in increments of 0.1. Within each experiment, we perform 10 runs, each with a different input schedule. We designate two types of delivery schedules, long and short, which are distinguished by the lengths of their sessions. Long schedules contain sessions twice as long, but half as many compared to those in a corresponding short schedule. Thus, the total task demands in S_O are equal for long and short schedules. Task demand per timestep is fixed at 25 units with each session in S_O.

We define Stress_index as a measure of agent resources relative to task demands. It is inversely proportional to Popsize because a smaller swarm will be under higher stress than a large one for constant task demands. It is directly proportional to the Scaling_factor because higher scaling effectively increases response thresholds, thus decreasing the number of agents that activate for a task.

Table 1 outlines experimental parameters, which are performed for the response threshold distributions and response duration values above. We examine experimental outcomes by looking at independent categorical variables (desynchronization method, schedule type), and the effects of *stress index* on interaction and within group variance for these variables. A typical way of analyzing a study with multiple variables, is to utilize a two-way repeated analysis of variance. Analysis of variance, or ANOVA, is used to study multiple categorical treatment means in an experiment, and their statistical significance with respect to results. In a two-way ANOVA, we can find a factor's influence on performance, just as with an ordinary ANOVA. However, we yield the added benefit of experimental interaction between groups of variables in our study. To analyze a factor's influence on performance, we must examine the p-value for a given experiment. If $p \leq .05$, we reject our null hypothesis. In this case, the null hypothesis, H_o, represents treatment means within a group being identical ($\mu_1 = \mu_2 = ... = \mu_N$). As an example, for a given experiment, we may find that no significant difference in mean performance between each threshold distribution, as a result, H_o, should be accepted. If any of the distributions contributed a significant change in performance, H_o should be rejected. If an interaction effect exists between two factor levels, this means that the paired combination is leading to a significant difference in outcomes.

Confirmation of Individual Effects: Here, we provide an analysis of desynchronization in terms of primary measures of success for our system.

1. *Response Threshold*: With each experiment, we have tested 10 runs for each schedule type, on each of the threshold distributions. First we observe, exclusively, the patterns in performance categories for each distribution and schedule type, independent of experimental parameters such as population size, and scaling value. Rather unsurprisingly, an underlying trade-off observed in response threshold results is found between performance in terms of speed (time to completion) and efficiency. Cost is often associated with wasted effort (or supply), and excessive task switching. Overall, uniform distributions, along with Poisson, tend to strike an ideal balance between these two metrics of performance.

2. *Response Duration*: We tested response duration on the range of values (0, 1] in intervals of 0.1. As the `Prob_check` value increases we observe a higher proportion of active agents and can satisfy task demands quicker. However, because we have more agents that are active, we tend to overshoot the required material demands. Alternatively, we observe an upward trend in task-switching as we increase `Prob_check`. With a longer duration of response, agents aren't constantly reevaluating their demands, and will only switch less often as a direct result. We find, for all 3 metrics, that short schedules perform worse. This was especially the case for overshooting quantities. We end up supplying more than necessary, the longer we act on sessions that are likely to end during the duration of response. This happens more often when sessions are shorter in length.

Broad Analysis of Swarm Behavior: System performance metrics are used for our Two-way repeated ANOVA. The experimental design is motivated by an assessment of the main effects given by both categorical variables, and their interaction. Experimental p-values are consolidated into a single table, for each metric. Sufficiently small p-values ($p \leq 0.05$) signify rejection of a null hypothesis, and thus, a statistically significant effect for the given parameter.

Table 2 shows two-way ANOVA results for timesteps to schedule completion averages for response threshold experiments. The first two columns show the experiment and corresponding stress index measure. The three remaining columns display p-values for the associated main effects, and interaction effect. For tables 2 and 3, we examine threshold and schedule type as independent variables for each experiment. From table 2 we observe, from experiment A to C, as stress decreases via scaling factor, interaction between threshold and schedule type gets weaker. We stop rejecting the null hypothesis once we descend to a scaling factor of 25. Increasing population size, from experiments D to G, lacks the same effect for response threshold experiments. Likewise, table 3 shows similar results for task-switching averages. As scaling values decrease from experiments A to C, we see the same trend occurring with the interaction effect results found in the fifth column, only this time, we stop rejecting the null hypothesis, H_o, at a stress index of 0.50.

Table 2: Two-way ANOVA PR(>F) for timesteps to completion averages on response threshold experiments.

Expt	Stress index	Main effect: Init_thresh	Main effect: Sched_type	Interaction effect
A	1.00	6.01E-108	7.93E-14	2.20E-08
B	0.50	4.46E-89	1.03E-26	5.03E-03
C	0.25	5.56E-60	8.06E-31	3.55E-01
D	2.00	7.66E-83	3.51E-12	2.97E-10
A	1.00	6.01E-108	7.93E-14	2.20E-08
E	0.67	1.22E-119	3.21E-24	2.39E-14
F	0.50	8.18E-39	2.07E-17	1.11E-25
G	0.25	2.07E-17	3.73E-04	1.12E-04

Table 3: Two-way ANOVA PR(>F) for agent task-switching averages on response threshold experiments.

Expt	Stress index	Main effect: Init_thresh	Main effect: Sched_type	Interaction effect
A	1.00	1.05E-58	3.67E-07	1.64E-18
B	0.50	5.60E-16	1.00E+00	9.99E-01
C	0.25	7.89E-01	5.90E-01	9.43E-01

Tables 2 and 3 both show null hypothesis rejection for interaction effects more often when threshold scaling values are higher. This is due to a wider spread of values across our range of possible response thresholds. Conversely, when scaling values are lowered, more homogeneity appears in agent performance, spanning multiple factors. When this is the case, changing threshold quantities or distributions clearly has a greater burden or weighted impact on performance - especially in the case of timesteps metrics. Figure 2 contains a set of interaction plots for timesteps results from experiments A and B. Along the x-axis, each threshold distribution is shown. The y-axis represents the average timesteps for schedule completion in this experiment for each setting. Both plots contain separate lines for schedule types. Reflected here are the findings from table 2. Comparing experiments A and B, we see a sharp decline in interaction between threshold and schedule type. Experiment B, having a stress index that is a half of experiment A, corroborates this relationship between stress and interaction within the grouping of independent variables. Figure 3 includes similar interaction plots as shown in figure 2, for task-switching results from experiments A and C. Here, we observe a decrease in interaction from experiment A to C, via decreasing scaling factor. Table 3 reveals the same trend with respect to interacting variables in response to these changes. Additionally, we find both main effects, in isolation, to be statistically similar across factor levels, accepting the null hypothesis for all p-values in experiment C, when stress is decreased. In general, placing higher demands on the swarm leads to greater interaction between parameters. When threshold scaling factors are larger, the load on our

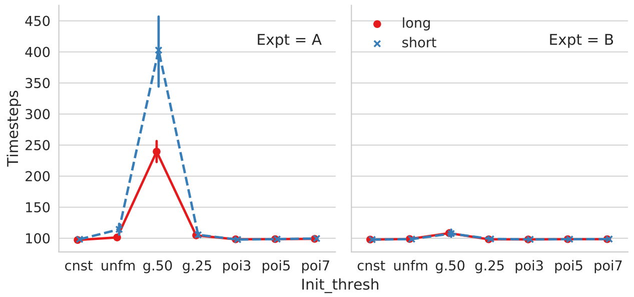

Figure 2: Interaction plots depicting average timesteps performance for threshold distribution and schedule type combinations, comparing experiments A and B, with scaling factors 100 and 50, respectively.

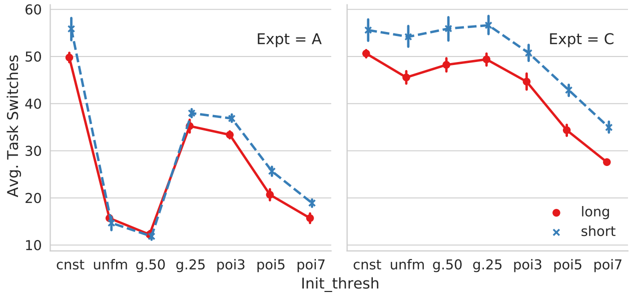

Figure 3: Interaction plots depicting average task switching for threshold distribution and schedule type combination, comparing experiments A and C, with scaling factors 100 and 25, respectively.

agents is intensified, since they cannot detect incoming task demands as quickly. Population size did not share this quality, despite its inverse relationship with stress index.

Table 4 shows Two-way ANOVA timesteps results for response duration, with a decreasing `Stress_index` by way of increasing `Popsize`. As with tables 2 and 3, we display experiment and stress index in the first two columns, along with p-values in the remaining columns for each factor. Observe that we reject H_o in the interaction effect column for higher stress indices from experiments D, A, and E. Note, again, that we find accepted null hypotheses for the remaining experiments F and G, for which there are lower stress index values. Across these experiments, stress is decreased via increasing population size. Higher population sizes play a role in diminishing the effects of changing response duration. Figure 4, similar to figures 2 and 3, shows the corresponding interaction plot with each response duration level. We add a third schedule type, `ushort`, with half the number of sessions of `short`. With an overabundance of agents, task demands are almost always satisfied. In this case, we base our analysis on changes to task switching due to increasing population. In comparison with response threshold results, it is clear that population size has a greater impact on duration of response, independent of threshold distribution.

Table 4: Two-way ANOVA PR(>F) for timesteps elapsed averages on response duration experiments.

Expt	Stress index	Main effect: Prob_check	Main effect: Sched_type	Interaction effect
D	2.00	8.66E-257	1.07E-121	1.08E-105
A	1.00	1.23E-168	6.84E-08	1.92E-13
E	0.67	3.15E-163	1.86E-05	4.61E-02
F	0.50	8.35E-144	1.03E-04	4.09E-01
G	0.25	2.89E-148	1.44E-15	7.73E-01

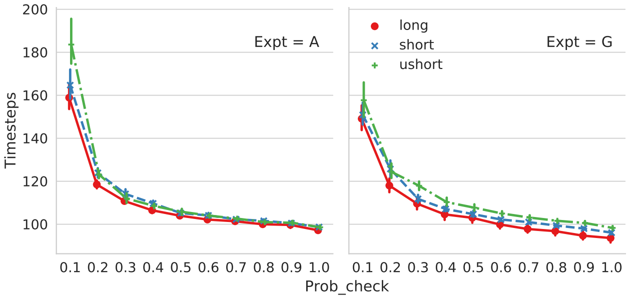

Figure 4: Interaction plots depicting average timesteps performance for different levels of response duration, comparing experiments A and G, with population sizes 100 and 400, respectively.

Since duration experiments only include agents with constant thresholds of zero, all agents will consider any active session for selection, since any active session automatically meets every threshold. As a direct consequence, of threshold distribution controlling what percentage of agents may act, we observe that population size, as a stressor, has less impact on threshold-schedule interaction.

Discussion

Agent performance within our system is defined by responsiveness (timesteps to completion), and efficiency (taskswitching and overshooting). Here, we have tested the effect of agent desynchronization on system-wide performance. We look specifically at how outcomes are affected by factors in a two-way experiment: our input schedule and a factor level associated with our mode of desynchronization. Additionally, we analyze the influence of swarm-based parameters, population size, and a threshold scaling factor. Both of these variables contribute to the relative amount of *stress* that our system experiences.

An examination of the two-way analysis reveals that interaction between threshold distribution and schedule variants diminishes under low-stress environments. Certainly, our resupply problem becomes more and more trivial, as we have an increasing supply of agents to deliver resources. As the problem gets harder with short schedules, we observe that agent performance suffers. When we overwhelm our agents, we only have idle agents when there are no active sessions. When we ensure that agents can adequately balance a schedules incoming task demands, we see an increase in the number of idle agents during a schedule's run. We find that both mechanisms for desynchronization impact swarm behavior, but can have different effects. Variable response duration diversifies the frequency with which agents re-evaluate their actions and affects how quickly agents respond to changing task demands, and variable response thresholds cause different agents to potentially react differently the same material, meaning that agents will be desynchronized in their acceptance of tasks for any given material to be selected.

References

Agassounon, W., and Martinoli, A. 2002. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proc. Int'l Conf. Autonomous Agents and Multi-Agent Systems*, 1090–1097.

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

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 IEEE International Conf. on Control, Automation, and Systems*, 259–266.

Castello, E.; Yamamoto, T.; Libera, F. D.; Liu, W.; Winfield, A. F. T.; Nakamura, Y.; and Ishiguro, H. 2018. Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach. *Swarm Intelligence* 10:1–31.

Correll, N. 2008. Parameter estimation and optimal control of swarm-robotic systems: A case study in distributed task allocation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 3302–3307.

de Lope, J.; Maravall, D.; and Quinonez, Y. 2015. Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems. *Neurocomputing* 163:47–55.

dos Santos, F., and Bazzan, A. L. C. 2009. An ant based algorithm for task allocation in large-scale and dynamic multiagent scenarios. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 73–80.

Engholdt, K.; Mathias, H. D.; and Wu, A. S. 2020. Variable response duration promotes self-organization in decentralized swarms. In *Proc. 9th Int'l Conf. on Bioinspired Optimization Methods and Their Applications*, 17–28.

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

Jones, C., and Mataric, M. J. 2003. Adaptive division of labor in large-scale minimalist multi-robot systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1969–1974.

Jones, J. C.; Myerscough, M. R.; Graham, S.; and Oldroyd, B. P. 2004. Honey bee nest thermoregulation: Diversity promotes stability. *Science* 305(5682):402–404.

Kanakia, A.; Touri, B.; and Correll, N. 2016. Modeling multi-robot task allocation with limited information as global game. *Swarm Intelligence* 10:147–160.

Kazakova, V. A., and Wu, A. S. 2018. Specialization vs. re-specialization: Effects of Hebbian learning in a dynamic environment. In *Proc. 31st Int'l Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 354–359.

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.

Liu, W.; Winfield, A.; Sa, J.; Chen, J.; and Dou, L. 2007. Towards energy optimisation: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior* 15:289–305.

Meyer, B.; Weidenmuller, A.; Chen, R.; and Garcia, J. 2015. Collective homeostatis and time-resolved models of self-organised task allocation. In *Proceedings of the 9th EIA International Conference on Bio-inspired Information and Communication Technologies*, 469–478.

Niccolini, M.; Innocenti, M.; and Pollini, L. 2010. Multiple UAV task assignment using descriptor functions. In *Proc. 18th IFAC Symp. Automatic Control in Aerospace*, 93–98.

Nouyan, S.; Ghizzioli, R.; Birattari, M.; and Dorigo, M. 2005. An insect-based algorithm for the dynamic task allocation problem. Technical report, IRIDIA. TR/IRIDIA/2005-031.

Pang, B.; Zhang, C.; Song, Y.; and Wang, H. 2017. Self-organized task allocation in swarm robotics foraging based on dynimcal response threshold approach. In *Proc. 18th International Conference on Advanced Robotics*, 256–261.

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

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

Weidenmüller, A.; Chen, R.; and Meyer, B. 2019. Reconsidering response threshold models—short-term response patterns in thermoregulating bumblebees. *Behavioral Ecology and Sociobiology* 73(8):1–13.

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

Wu, A. S., and Mathias, H. D. 2020. Dynamic response thresholds: Heterogeneous ranges allow specialization while mitigating convergence to sink states. In *Proc. 12th International Conf. Swarm Intelligence*, 107–120.

Wu, A. S.; Mathias, H. D.; Giordano, J. P.; and Hevia, A. 2020. Effects of response threshold distribution on dynamic division of labor in decentralized swarms. In *Proc. 33rd FLAIRS*, 386–391.

Yang, Y.; Chen, X.; and Li, Q. 2010. Swarm robots task allocation based on local communication. In *Proc. International Conference on Computer, Mechatronics, Control, and Electronic Engineering*, 415–418.

Yasuda, T.; Kage, K.; and Ohkura, K. 2014. Response threshold-based task allocation in a reinforcement learning robotic swarm. In *Proc. 7th IEEE Int'l Workshop on Computational Intelligence and Applications*, 189–194.