Variable response duration promotes self-organization in decentralized swarms*

Kaelan Engholdt[1], H. David Mathias[1], and Annie S. Wu[2]

[1] University of Wisconsin – La Crosse, La Crosse, WI 54601, USA, {engholdt8911, dmathias}@uwlax.edu

[2] University of Central Florida, Orlando, FL 32816, USA, aswu@cs.ucf.edu

Abstract. In self-organizing multi-agent systems, inter-agent variation is known to improve swarm performance significantly. Response duration, the amount of time that an agent spends on a task, has been proposed as a form of inter-agent variation that may be beneficial. In the biological literature, variability in agent response duration in natural swarms for desynchronizing agent actions has been discussed for some time. This form of variation, however, is not well understood in artificial swarms. In this work, we explore inter-agent variation in response duration as a desynchronization technique. We find that variation in response duration does desynchronize agent behaviors and does improve swarm performance on a two-dimensional tracking problem in which the swarm must push a tracker, staying as close as possible to a moving target. By preventing agents from reacting identically to task stimuli and keeping some agents on task longer, response duration helps smooth the swarm's path and allows it to better track the target into path features such as corners.

Keywords: Multi-agent system · Inter-agent variation · Response duration · Response threshold.

1 Introduction

In this paper, we investigate variable response duration as a mechanism for promoting effective self-organization in a decentralized swarm. The decentralized and redundant structure of swarms make them potentially very robust and adaptable. These same qualities, however, also make the task of coordinating agents within a swarm a challenging problem. For a swarm to address any reasonably interesting problem, the agents in the swarm must be able to distribute themselves intelligently among multiple tasks, even in problems where task stimuli are globally sensed by all agents. Such coordinated responses can be difficult to achieve when all agents act independently. A significant body of work has studied how variation in *when* agents are triggered to act, i.e. variation in response threshold, can desynchronize agent actions enough to generate effective

* Supported by NSF Grant IIS1816777.

division of labor [6, 15, 16, 20, 21, 24, 23]. Biological studies hypothesize that variation in *how long* agents work before they stop to re-assess their actions can also contribute to desynchronizing decentralized swarms of agents [22]. This work examines the effectiveness of variable response duration on swarm self-organization and its strengths and weaknesses relative to and in conjunction with variable response thresholds.

Effective self-organization of decentralized swarms requires desynchronization of the agents within a swarm to achieve diversity in agent actions. If agents make action decisions at different times, they are likely to encounter different stimuli and, thus, have the potential to act differently. A commonly used method of desynchronization is variation in response threshold. Giving each agent a different threshold for each task stimulus causes agents to be triggered at different times by a given stimulus. As a result, agents enter the workforce gradually rather than all at the same time and entry into the workforce may stop after task needs have been fully addressed by a subset of agents. Simply assigning agents randomly generated thresholds over a uniform distribution is sufficient to generate division of labor in a swarm [15, 23]. Studies have examined both static distributions of thresholds [8, 12, 15, 18, 19, 23, 24] and dynamically evolved thresholds [2–7, 9, 10, 13, 14, 19, 21].

Weidenmuller's [22] study on honeybee thermoregulation points out that a different factor may also contribute to the desynchronization of agent actions in a decentralized swarm: variation in response duration. Response duration refers to the amount of time that an agent works on a task before stopping to re-assess its actions. Instead of all agents evaluating task demands and selecting an action in every unit of time, agents may work differing numbers of time units on a task before stopping to reconsider task demands. The varying durations cause agents to be desynchronized with respect to when they evaluate task stimuli, increasing the chance that they will sense different stimuli and react diversely.

Active adjustment of the amount of time agents spend on tasks is not new to swarm self-organization studies. Factors such as the amount of time that agents have been resting or active [1, 17], agent success rate or productivity on task [25], and perceived relative task demands [11] have been used to affect when agents start and stop work on tasks. In all of these approaches, however, response durations are tied in part to external forces, e.g. the availability of jobs or density of jobs. As such, even though agents act independently, there exists the possibility that the external forces that are driving their response durations could inadvertently synchronize agents.

We are interested in response duration as an inherent characteristic of an agent and whether variation in agent response duration within a swarm can contribute to more effective self-organization. We study the performance of a decentralized swarm on a collective tracking problem. We first examine whether the desynchronizing effects of variable response duration is able to improve a swarm's ability to self-organize. We then explore the implications of varying the average expected response duration lengths of the agents. Finally we present interesting results combining two mechanisms for desynchronizing decentralized

swarms: variable response durations combined with variable response thresholds. Our results indicate that variable response duration is a viable method for improving self-organization in swarms and suggest that the interaction of multiple forms of inter-agent variation may result in richer behavior than any single form alone.

2 Problem description

In seminal work on inter-individual variation in bumblebees [22], Weidenmuller explores collective nest thermoregulation. In that problem, individual bees choose between two tasks, flapping their wings or shivering, to lower or raise the temperature in the hive, respectively. Thermoregulation is a one-dimensional problem in which the two tasks are in opposition. For the testbed in this work, we use a two-dimensional tracking problem in which the swarm attempts to move an object to track, as closely as possible, a target. The target's path is unknown to the agents comprising the swarm. Superficially very different from thermoregulation, this problem is quite similar though more complex.

A simulation is divided into a predetermined number of time steps. During each time step, the target moves a fixed distance in a direction determined by the underlying path. Random paths may change direction as often as every time step, creating frequently changing task demands, while periodic paths create periods of nearly constant task demands followed by brief periods of abrupt changes.

To track the target, each agent can undertake one of four tasks in each timestep: `push_NORTH`, `push_SOUTH`, `push_EAST`, or `push_WEST`. Task demands for the swarm are determined by movement of the target. Agents are aware of the demands in the form of the task stimuli, the distances between the target and tracker in each dimension. Let $\Delta x = \text{target}.x - \text{tracker}.x$ and $\Delta y = \text{target}.y - \text{tracker}.y$. Task stimuli are defined as: $\sigma_N = -\Delta y$, $\sigma_E = -\Delta x$, $\sigma_S = \Delta y$, and $\sigma_W = \Delta x$.

Whether an agent acts in a given time step is determined by the task stimuli and one or more forms of inter-agent variation described previously. Without inter-agent variation, agents respond in lockstep to stimuli, inhibiting the ability of the swarm to perform a variety of tasks.

Performance of a swarm is measured relative to the following two domain goals:

Domain Goal 1 *Minimize the average positional difference, per time step, between the target location and the tracker location.*

Domain Goal 2 *Minimize the difference between total distance traveled by target and the total distance traveled by the tracker.*

We note that neither criterion alone is sufficient to gauge the swarm's success. Consider using only Goal 1. The tracker could remain close to the target while alternately racing ahead or falling behind. This would result in a good average

difference but a path length that is significantly greater than that traveled by the target. Alternately, using only Goal 2, the tracker could travel a path that is the same length as that of the tracker while straying quite far, taking a very different path.

Both the honeybee thermoregulation problem and the tracking problem are examples of decentralized task allocation problems. There are certainly more effective methods to achieve tracking and the focus of this work is not on that problem domain. Rather the tracking problem is used here because it is an example of a decentralized task allocation problem in which task demands and contributions are clearly defined and measured, dynamic variation in task demand over time can be systematically described, and overall performance can be accurately measured as well as visually assessed.

3 Experimental details

To examine the effect of response duration, we vary the time period for which an agent performs a task. This is done via a parameter named `Prob_check`. This parameter represents the probability that in any time step an agent will undergo task selection. If an agent does not undergo task selection, it continues working on its current task. It is important to note that `Prob_check` is inversely proportional to response duration. That is, a high `Prob_check` value results in less time spent on a task (more frequent task selection) while low `Prob_check` values result in more time spent on a task (less frequent task selection). We perform experiments with `Prob_check` values in $[0.1, 1.0]$ in increments of 0.1.

We perform experiments using two target paths, circle and serpentine.

- circle: Target continuously revolves about a central point at a fixed distance, resulting in a circular path with radius r. This creates continuously changing task demands and requires the swarm to perform all tasks equally.
- serpentine: A periodic path that oscillates up and down, moving from west to east. The motion is defined by amplitude and period values. `Path_amplitude` dictates how far the target moves in the north and south directions. `Path_period` controls the distance between peaks in the waveform.

The bottom of Table 1 shows parameters that allow some variation in the circle and serpentine paths. These include seven radii for circle and four pairs of amplitudes and periods for serpentine. With these parameter values, we can affect the rate at which task demands change.

The top of Table 1 lists parameters that are fixed for all experiments. These include the number of agents in the swarm and the number of time steps in each simulation. Two other parameters require some explanation. When selecting a task, agents may choose `push_NORTH`, `push_SOUTH`, `push_EAST`, or `push_WEST`. Urgent task selection means that agents will select the task with the greatest task demand. In each time step, the target moves a fixed distance defined by `Target_step_len`. This value is fixed at 3 for these experiments. The maximum distance the tracker can move is defined by the `Target_step_len` times the

Table 1. Parameters with values fixed across all experiments (top) and those with values that vary by experiment (bottom).

Parameter	Value
Population size	200
Time steps	500
Task selection	urgent
Target_step_len	3
Step_ratio	1.5
Prob_check	[0.1, 1.0] by 0.1
Radius (circle)	3, 5, 10, 15, 20, 25, 30
Path_amplitude (serpentine)	6, 9, 12, 15
Path_period (serpentine)	10, 20, 30, 40

`Step_ratio`. With a `Step_ratio` of 1.0, all agents would have to push in the right direction in order for the tracker to keep up with the target. Higher `Step_ratio` values allow for some agents to remain idle or undertake a wrong task without severe consequences for the swarm.

Each experiment consists of 100 runs. We average data across all runs and calculate 95% confidence intervals. We measure swarm performance by two data, average positional difference between the target and tracker (Goal 1) and difference between target and tracker path lengths (Goal 2).

- Average Positional Difference: This is the mean over all time steps of the Euclidean distance between the target and the tracker positions. This measures the deviation between the target and tracker paths over the course of the run. The optimum value for this metric is zero.
- Path Length Difference: This is a measure of the difference between the total path lengths traveled by the tracker and the target. A negative value for this metric indicates that the tracker did not travel as far as the target, where as a positive value means the tracker traveled a longer path than did the target. The optimum value for this metric is zero.

4 Results

4.1 Can variable response duration improve self-organization?

We begin by asking the general question: does the desynchronization of agents that results from variable response duration improve a swarm's ability to self-organize? Figure 1 shows two example instances of a tracker's attempt to follow a target along a circular path and a serpentine path. The left column shows the results for `Prob_check` = 1.0, which is uniform response duration. The right column shows the results for `Prob_check` = 0.4, where each agent has a 40% chance of re-assessing its current action in each timestep and a 60% chance of ignoring current task demands and continuing with its current task. The top row shows a circular path. The bottom row shows a serpentine path.

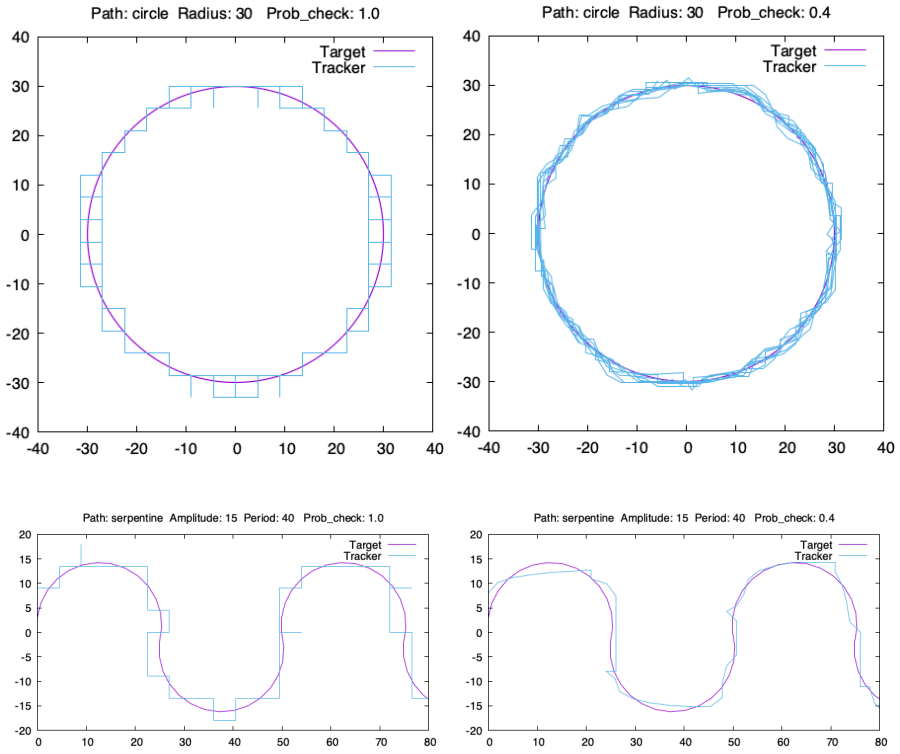

Fig. 1. Examples of tracker paths with `Prob_check` = 1.0 (left column) and `Prob_check` = 0.4 (right column) following target on a circular path (top row) and serpentine path (bottom row).

The plots in the left column show that, without any other mechanisms to diversify agent actions, all agents act identically in every timestep because they all perceive the same task demands in each timestep. As a result, the tracker only moves in the four cardinal directions and the path it traces as it follows the target is very blocky. The plots in the right column provide evidence that variable response duration can have a desynchronizing effect that diversifies agent actions. The `Prob_check` value less than 1.0 causes only a subset of agents to re-assess their actions in each timestep. Even though all of those agents may select the same task, the swarm as a whole has a diversity of agent actions in each timestep because of the agents that maintain their current task. As a result, the tracker is able to move in more than just the four basic directions and is able to follow the path of the target more closely and trace a smoother path. Thus, the desynchronization effect of variable response duration can produce diversity in agent actions that can improve self-organization.

4.2 Implications of variable response duration

While variable response duration appears to desynchronize agent actions, it also causes the swarm to react less promptly to task demands as swarms in which all agents re-assess their actions in every timestep. As a result, the desynchronization benefits of variable response duration may come at a cost of lowered swarm responsiveness to changing task demands. That suggests that a problem with frequently changing task demands will be more difficult for smaller `Prob_check` values (longer durations) than larger `Prob_check` values (shorter durations). We explore this hypothesis by examining the full range of `Prob_check` values and how they respond to paths that require increasing levels of responsiveness.

For a given `Prob_check` value d, we can estimate the expected duration that agents will stay on a task before re-assessing its actions. Let τ be the number of time steps that an agent performs a task. For `Prob_check` value d, the expected value of τ is given by $E(\tau) = 1/d$. We then calculate the average number of time steps that agents act for each `Prob_check` value in an example run of the tracking simulation. Table 2 shows that the empirically observed durations from a sample tracking run closely match the expected ($E(\tau)$) value for all of the `Prob_check` values.

Table 2. Expected and observed number of time steps for an agent with a given `Prob_check` value to consider changing tasks.

`Prob_check`	d	$E(\tau)$	Observed duration	`Prob_check`	d	$E(\tau)$	Observed duration
0.1		10.00	10.150221	0.6		1.67	1.664558
0.2		5.00	5.003253	0.7		1.43	1.431783
0.3		3.33	3.335890	0.8		1.25	1.249157
0.4		2.50	2.500249	0.9		1.11	1.111148
0.5		2.00	1.998521	1.0		1.00	1.000000

Figure 2 shows the Path Length Difference and Average Positional Difference measures averaged over 100 simulation runs for `Prob_check` values from 0.1 to 1.0 on circle paths with radii ranging from 30 down to 3. The x-axes of each plot indicates `Prob_check` values. The y-axes of the left column of plots indicate Path Length Difference. The y-axes of the right column of plots indicate Average Positional Difference. Each row of plots gives the results for a circular target path with radii ranging from 30 down to 3. As a target path's radius decreases, required swarm responsiveness is expected to increase and we expect higher `Prob_check` values to be needed to maintain performance.

The left column of Figure 2 shows the average Path Length Difference averaged over 100 runs with 95% confidence interval. Recall that the optimal Path Length Difference is zero; a positive value indicates that the tracker travels farther than the target; a negative value indicates that the tracker travels a shorter distance than the target. Looking at the top plot in the left column of Figure 2, we see that, on a circle of radius 30, tracker and target path lengths are most similar at the lowest `Prob_check` values. As `Prob_check` values increase, the tracker

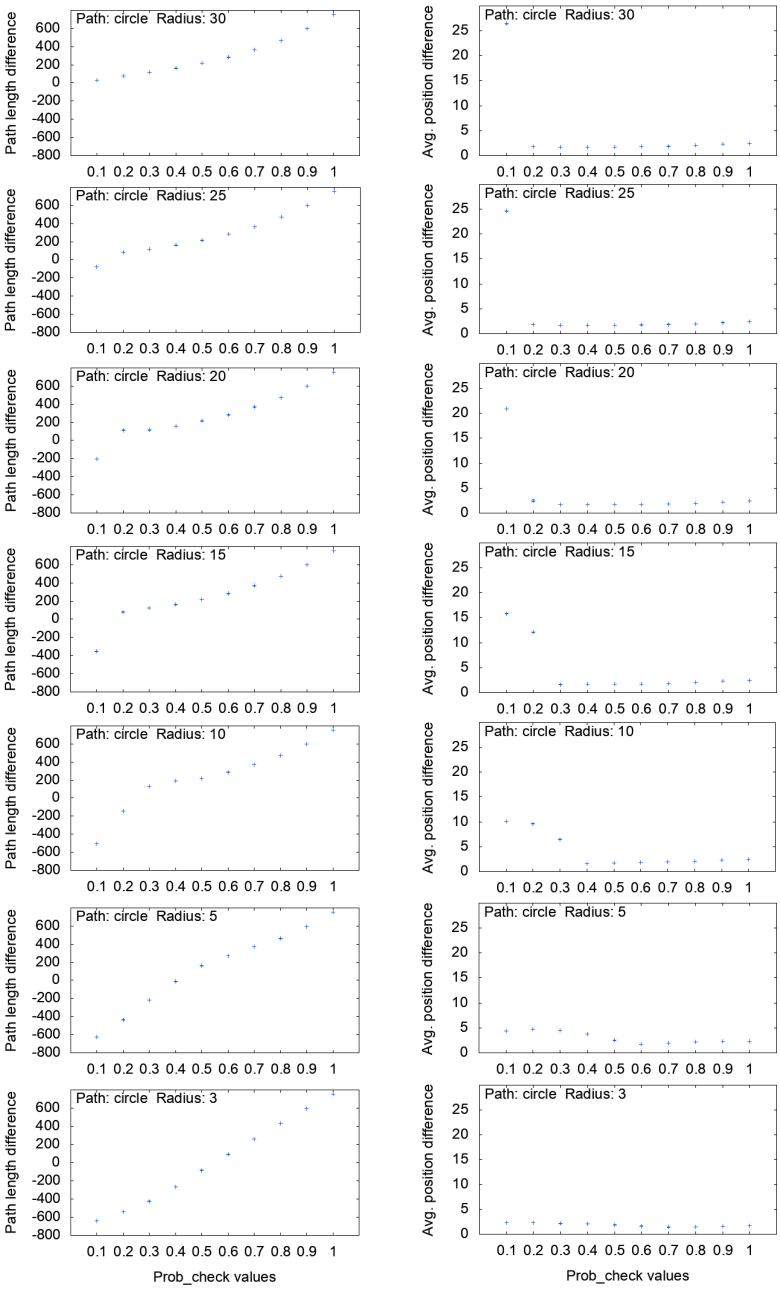

Fig. 2. Path Length Difference (left column) and Average Positional Difference (right column) measures averaged over 100 runs with 95% confidence intervals. Rows show results from circles with path radii from 30 to 3.

travels increasingly longer distances relative to the target. These extra distances are due to the overshooting that occurs when too many agents respond, as seen in the blocky movements of the left plot in Figure 1. As we move down the rows in Figure 2, the plots show data from smaller and smaller circles. As the circle radii shrink, the optimal Path Length Difference values are found at increasingly higher `Prob_check` values. At radius 10, optimal `Prob_check` is between 0.2 and 0.3; at radius 3, optimal `Prob_check` is between 0.5 and 0.6. Also as the circle radii shrink, lower `Prob_check` values start producing more and more negative Path Length Difference values due to swarm responsiveness being too low for the tracker to keep up with the target. Thus, as the target path (and task demands) change more quickly, higher `Prob_check` values are needed to achieve optimal performance because agents need to re-assess and adjust their actions more frequently to keep up with the target movement.

The right column of Figure 2 shows the Average Positional Difference averaged over 100 runs with 95% confidence interval. The optimal Average Positional Difference is zero. The top right plot of Figure 2 shows that all `Prob_check` values except for `Prob_check` = 0.1 achieve Average Positional Difference of approximately 2 or less. As we move down the rows in Figure 2, `Prob_check` = 0.1 results improve but still lag behind higher `Prob_check` values. The change in the results for `Prob_check` = 0.2 and 0.3 as circle radius decreases is more interesting. At high circle radii, these values are relatively low. At radii of 15 and 10, the Average Positional Difference for `Prob_check` = 0.2 and 0.3, respectively, increase significantly. As radii continue to decrease, the results for both slowly drop back down to values around 2. Higher `Prob_check` values consistently achieve low Average Positional Difference values.

Examination of individual runs explains these observed results as follows. At high circle radii, the change in task demand is low from one time step to the next. As a result, even trackers with relatively low `Prob_check` values can follow the target path. As circle radii decrease, the change in task demand from one time step to the next increases. The increased change requires a more responsive swarm for the tracker to be able to keep up with the target. When the target task demand changes become too great for given `Prob_check` value to keep up, we see an increase in the Average Positional Difference. At this point, the tracker lags so far behind the target that significant corner-cutting occurs. Because the example runs presented are on a circular path, the corner cutting by the tracker leads it to travel in a circular path within the target's circular path. As the circle radii continue to decrease, the change in task demand becomes more frequent. Although the swarm has difficulty keeping up with the target, the decreasing radii results in lower Average Positional Difference because the tracker is constantly moving within the target path due to corner cutting. This behavior is not unique to the circular target path; similar degradation trends are observed in other path results.

4.3 Combining response duration and response thresholds

While the results above show that variation in response duration can benefit a swarm's ability to self-organize, they also suggest that the effectiveness of this mechanism may depend on a good pairing of `Prob_check` value with the dynamism of the path being tracked. In other words, optimal use of response duration is only possible with a priori knowledge of the problem to which a swarm is applied. Because a priori information is not always available, we explore other situations where variation in response duration may be generally beneficial.

Specifically, we find that combining variable response durations with variable response thresholds can provide added benefits. Variable response thresholds have been shown to be a successful method for desynchronization of decentralized agents. When agent resources are insufficient for addressing all problem demands, however, such systems will fall behind and task attendance may lag task demand. In the tracking problem such lags often manifest as corner cutting. Empirical studies show that the lowered responsiveness that emerges from longer response durations often results in a delayed reaction that resembles "inertia" in agent task choices. Figure 3 shows an example of a serpentine path with a swarm using variable response thresholds alone (left) and a swarm using variable response thresholds combined with variable response duration (right). While the path on

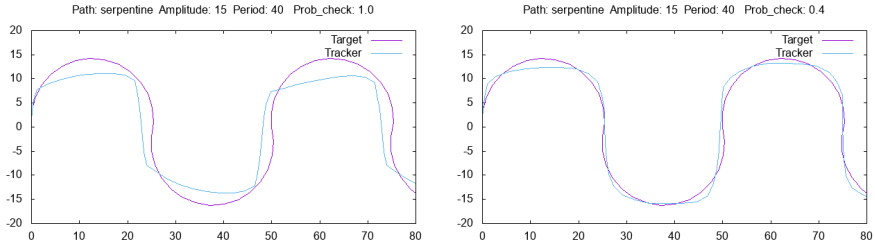

Fig. 3. Effects of combining variable response duration with variable response thresholds. Left: Variable response thresholds alone results in corner cutting. Right: When variable response thresholds is combined with variable response duration, the delayed reaction resulting from longer response durations mitigates corner cutting.

the left cuts corners, in the path on the right, the delayed response generated by variable response duration pushes the tracker further into each turn and reduces the corner cutting effect. Thus, while both variable response thresholds and variable response durations may be used to desynchronize the actions of decentralized agents, they do so using different mechanisms and there appear to be potential advantages to combining multiple mechanisms.

5 Conclusions

In this work, we examine the impact of variable response duration on the ability of a decentralized swarm to self-organize. We demonstrate that inter-agent variation in response duration serves to desynchronize agent actions, improving self-organization. A side effect of variable response duration is delayed response to changing task demands when compared to swarms in which all agents undergo task selection in every timestep. We can mitigate this effect with knowledge of the target path. Tracking of paths with more frequent changes in task demands benefits from higher `prob_check` values (more frequent task selection).

A priori knowledge of a problem is not always possible and we may not be able to choose effective `prob_check` values. Thus, we combine variation in response duration with variation in response thresholds. We show that these forms of inter-agent variation are complementary due to using different mechanisms for desynchronization. The delayed response due to response duration counters the tendency to cut corners caused by use of response thresholds alone. Thus, for the tracking problem, this combination appears to be beneficial.

One limitation of this work is that we consider only a single problem domain. In future work, we plan to use additional problems to test the utility of response duration. In addition, we will explore combinations of response duration with other forms of inter-agent variation to determine if similar synergies exist.

References

1. Agassounon, W., Martinoli, A.: Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems. pp. 1090–1097 (2002)
2. Campos, M., Bonabeau, E., Theraulaz, G., Deneubourg, J.: Dynamic scheduling and division of labor in social insects. Adaptive Behavior **8**, 83–96 (2000)
3. Castello, E., Yamamoto, T., Libera, F.D., Liu, W., Winfield, A.F.T., Nakamura, Y., Ishiguro, H.: Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach. Swarm Intelligence **10**, 1–31 (2018)
4. Castello, E., Yamamoto, T., Nakamura, Y., Ishiguro, H.: Task allocation for a robotic swarm based on an adaptive response threshold model. In: Proceedings of the 13th IEEE International Conference on Control, Automation, and Systems. pp. 259–266 (2013)
5. Cicirello, V.A., Smith, S.F.: Distributed coordination of resources via wasp-like agents. In: Lecture Notes in Artificial Intelligence. vol. 2564, pp. 71–80 (2002)
6. de Lope, J., Maravall, D., Quinonez, Y.: Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems. Robotics and Autonomous Systems **61**, 714–720 (2013)
7. de Lope, J., Maravall, D., Quinonez, Y.: Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems. Neurocomputing **163**, 47–55 (2015)
8. dos Santos, F., Bazzan, A.L.C.: An ant based algorithm for task allocation in large-scale and dynamic multiagent scenarios. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 73–80 (2009)

9. Gautrais, J., Theraulaz, G., Deneubourg, J., Anderson, C.: Emergent polyethism as a consequence of increase colony size in insect societies. Journal of Theoretical Biology **215**, 363–373 (2002)
10. Goldingay, H., van Mourik, J.: The effect of load on agent-based algorithms for distributed task allocation. Information Sciences **222**, 66–80 (2013)
11. Jones, C., Mataric, M.J.: Adaptive division of labor in large-scale minimalist multi-robot systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1969–1974 (2003)
12. Kanakia, A., Touri, B., Correll, N.: Modeling multi-robot task allocation with limited information as global game. Swarm Intelligence **10**, 147–160 (2016)
13. Kazakova, V.A., Wu, A.S.: Specialization vs. re-specialization: Effects of Hebbian learning in a dynamic environment. In: Proc. 31st Int'l Florida Artificial Intelligence Research Society Conference. pp. 354–359 (2018)
14. Kazakova, V.A., Wu, A.S., Sukthankar, G.R.: Respecializing swarms by forgetting reinforced thresholds. Swarm Intelligence (2020)
15. Krieger, M.J.B., Billeter, J.B.: The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. Robotics and Autonomous Systems **30**, 65–84 (2000)
16. Lee, W., Kim, D.: History-based response threshold model for division of labor in multi-agent systems. Sensors **17**, 1232 (2017)
17. Liu, W., Winfield, A., Sa, J., Chen, J., Dou, L.: Towards energy optimisation: Emergent task allocation in a swarm of foraging robots. Adaptive Behavior **15**, 289–305 (2007)
18. Meyer, B., Weidenmuller, A., Chen, R., Garcia, J.: 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. pp. 469–478 (2015)
19. Price, R., Tino, P.: Evaluation of adaptive nature inspired task allocation against alternalte decentralized multiagent strategies. In: In Proceedings of PPSN VIII. pp. 982–990 (2004)
20. Riggs, C., Wu, A.S.: Variation as an element in multi-agent control for target tracking. In: Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems. pp. 834–841 (2012)
21. Theraulaz, G., Bonabeau, E., Deneubourg, J.: Response threshold reinforcement and division of labour in insect societies. Proc. Royal Society B **265**, 327–332 (1998)
22. Weidenmüller, A.: The control of nest climate in bumblebee (*Bombus terrestris*) colonies: Interindividual variability and self reinforcement in fanning response. Behavioral Ecology **15**, 120–128 (2004)
23. Wu, A.S., Mathias, H.D., Giordano, J.P., Hevia, A.: Effects of response threshold distribution on dynamic division of labor in decentralized swarms. In: Proc. 33rd Int'l Florida Artificial Intelligence Research Society Conference (2020)
24. Wu, A.S., Riggs, C.: Inter-agent variation improves dynamic decentralized task allocation. In: Proc. 31st Int'l Florida Artificial Intelligence Research Society Conference. pp. 366–369 (2018)
25. Yasuda, T., Kage, K., Ohkura, K.: Response threshold-based task allocation in a reinforcement learning robotic swarm. In: Proceedings of the 7th IEEE International Workshop on Computational Intelligence and Applications. pp. 189–194 (2014)