Dynamic Response Thresholds: Heterogeneous Ranges Allow Specialization While Mitigating Convergence to Sink States

Annie S. Wu[1(✉)] and H. David Mathias[2]

[1] University of Central Florida, Orlando, FL, USA
aswu@cs.ucf.edu
[2] University of Wisconsin – La Crosse, La Crosse, WI, USA
dmathias@uwlax.edu

Abstract. We argue that heterogeneous threshold ranges allow agents in a decentralized swarm to effectively adapt thresholds in response to dynamic task demands while avoiding the pitfalls of positive feedback sinks. Dynamic response thresholds allow agents to dynamically evolve specializations which can improve the responsiveness and stability of a swarm. Dynamic thresholds that adapt in response to previous experience, however, are vulnerable to getting stuck in sink states due to the positive feedback nature of such systems. We show that heterogeneous threshold ranges result in comparable task allocation and improved stability as compared to homogeneous threshold ranges, and that simple static random thresholds should be considered in situations where agent resources are plentiful.

1 Introduction

In this paper, we show that heterogeneous threshold ranges allow agents in a decentralized swarm to effectively adapt thresholds in response to dynamic task demands while avoiding the pitfalls of positive feedback sinks. Response threshold based systems are a biologically inspired approach for generating division of labor in decentralized swarms [1,2,30]. While static thresholds are able to achieve effective task allocation [16,19,34], allowing agents to dynamically adapt their task thresholds over time allows for dynamic specialization which is thought to improve the responsiveness and stability of a swarm. Dynamic thresholds that adapt in response to previous experience, however, are vulnerable to getting stuck in sink states due to the positive feedback nature of such systems [17,30]. We show that varying the threshold ranges of each agent can effectively mitigate the negative effects of sinks while retaining the benefits of dynamic thresholds.

The response threshold approach is an effective method for generating task allocation in decentralized robotic swarms. Each agent possesses a threshold for each task that the agent can potentially take on. An agent's decision as to which

M. Dorigo et al. (Eds.): ANTS 2020, LNCS 12421, pp. 107–120, 2020.
https://doi.org/10.1007/978-3-030-60376-2_9

task, if any, to take on is a function of the agent's threshold for each task and the observed task stimuli. This approach is effective in decentralized systems and is not dependent on inter-agent communication which makes it scalable and useful for problems where stealth is necessary or where agents carry limited power. Response threshold approaches include both static [11,16,19,27,28,33,34] and dynamic thresholds [5–10,12,13,17,18,28,30]. Dynamic thresholds are particularly interesting because they allow a swarm to adjust the distribution of its agent propensities over time. For problems where the distribution of work is not known in advance or may change over time, this adaptability can potentially make the swarm more effective. In addition, dynamic thresholds allow agents to specialize on tasks which improves efficiency by reducing task switching [3,4].

In systems that use dynamic thresholds, agents may adjust their thresholds in response to external[1] [6,14,22–24] or internal factors [20,25,26]. The former tends to be problem dependent and out of the scope of this paper. We study the latter approach, specifically, systems modelled on the concept that previous experience on a task makes an agent more likely to act on that task in the future [12,21,29,30]. This concept is commonly implemented in the form of a learning factor that, in each timestep, lowers the threshold of a task on which an agent is working and a forgetting factor that increases that agent's threshold for all other tasks [5,8,28,30]. While agents in such systems can effectively converge their thresholds into a distribution that meets a given set of task demands, once converged, these systems often have difficulty undoing an expired distribution and re-adjusting to new demands if task demands change [17,18]. The positive feedback structure of this concept results in a tendency for thresholds to evolve to extreme values which are sink states that are difficult to subsequently evolve out of.

We hypothesize that heterogeneous threshold ranges can improve the performance of dynamic response threshold systems by reducing the effects of sink states while still allowing agents to adapt their thresholds and specialize on tasks. Current dynamic response threshold systems assign the same threshold range to all agents. This homogeneity means that once convergence occurs, all agents that have converged will be equally unwilling to revise their thresholds when task demands change. Heterogeneous threshold ranges would result in convergence to different values, allowing some agents to be more willing to revise their thresholds than others. In addition, should agents still get stuck in sink states, the variability in sink states may allow the swarm greater ability to respond to new task demands than if all agents are stuck in the same sink state.

2 Collective Tracking Problem

We test our hypothesis on a collective tracking problem [33,34] which attempts to model a collective task allocation problem similar to that of honeybee thermoregulation [15,31,32]. Where thermoregulation works in a single dimension

[1] External factors include but are not limited to task stimuli and observed actions of other agents.

with agents selecting from among two tasks, the tracking problem works in two dimensions with agents selecting from among four tasks.

The collective tracking problem consists of a target that moves in a two dimensional space and a tracker that is collectively controlled by the swarm. The goal of the swarm is to push the tracker such that its movement tracks the target as closely as possible. In each timestep, the individual agents in the swarm select from one of four tasks – *PUSH_NORTH*, *PUSH_EAST*, *PUSH_SOUTH*, *PUSH_WEST* – or remain idle. A positive difference between the target and tracker locations in any direction signifies a task demand in that direction. Each agent can select to push in, at most, one direction in each timestep. The tracker movement in each timestep is calculated by aggregating the decisions of all active agents in that timestep.

The path on which a target moves determines the task demands and how they change over time. For example, constant movement in the northeast direction results in constant equal task demands to the north and east in each timestep. A zigzag path represents task demands that remain stable for a period of time, but occasionally change significantly and abruptly. Serpentine or circular paths, on the other hand, represent constant gradual changes in task demands.

The authors acknowledge that there are more effective and efficient methods to accomplish tracking. We use this collective tracking problem as a testbed because it is a useful example of a decentralized task allocation problem. As the target moves through space, positive difference between the target and tracker in any direction represents a task demand in that direction. The relative number of agents that select to push in each direction determines the aggregate tracker movement; hence, accurate self-allocation of agents to tasks is required to meet task demands. The specification of a target path allows us a systematic way to define dynamic task demands with specific characteristics. The problem is designed such that we are able to quantitatively measure the satisfaction of each task demand individually as well as visually assess the overall performance of the system by comparing the actual target and tracker paths.

3 System Details

We compare the performance of a dynamic response threshold swarm using heterogeneous threshold ranges, termed Dynamic-Heterogeneous, against the performance of two baseline systems. The first baseline system, Dynamic-Homogeneous, is a dynamic response threshold swarm using homogeneous threshold ranges. Dynamic-Homogeneous is representative of how most current dynamic threshold systems work. The second baseline system, Static, is a swarm with static thresholds.

All three systems consist of a population of n decentralized agents, $a_i, i = 0, ..., n$. Each agent has a separate threshold for each task or direction, $\{\theta_{i,N}, \theta_{i,E}, \theta_{i,S}, \theta_{i,W}\}$. These thresholds represent the tolerance of that agent for the corresponding differences, $\{\Delta_N, \Delta_E, \Delta_S, \Delta_W\}$, between target and tracker position. In a given timestep, if the difference in a direction exceeds the agent's threshold for that direction (if $\Delta_j > \theta_{i,j}$), the agent will consider pushing in that direction for that timestep. If more than one task is triggered for an agent, the agent randomly selects one of the triggered tasks on which to act. Note that for this problem, because $\Delta_N = -\Delta_S$ and $\Delta_E = -\Delta_W$ are always true, not more than two tasks will ever be triggered at the same time.

Agent thresholds work as follows in the three systems tested. All thresholds are floating point values. In the two dynamic systems, each threshold, $\theta_{i,j}$, has a range within which it can vary. This range is defined by a minimum, $\theta_{i,j}min$, and maximum, $\theta_{i,j}max$, value. In the Static system, thresholds are static and are initialized uniformly randomly to a value between 0 and R, where R is a user specified parameter indicating the maximum allowed threshold value. In the Dynamic-Homogeneous system, thresholds are dynamic and all thresholds can vary within the range specified by $\theta_{i,j}min = 0$ and $\theta_{i,j}max = R$. The initial value of each threshold is a random value drawn from a uniform distribution between 0 and R. In the proposed Dynamic-Heterogeneous system, thresholds are dynamic and all thresholds vary within a unique range. The lower bound of the range, $\theta_{i,j}min$, is a random value drawn from a uniform distribution between 0 and $\frac{R}{2}$. The upper bound of the range, $\theta_{i,j}max$, is a random value drawn from a uniform distribution between $\frac{R}{2}$ and R. The initial value of each threshold is a random value drawn from a uniform distribution between $\theta_{i,j}min$ and $\theta_{i,j}max$.

For the two dynamic threshold systems, threshold variation occurs the same way as seen in previous work [30]. In each timestep, if an agent is working on a task j, its threshold for that task is decreased by a learning factor ε such that $\theta_{i,j} = \theta_{i,j} - \varepsilon$, and its thresholds for all other tasks are increased by a forgetting factor ψ such that $\theta_{i,j} = \theta_{i,j} + \psi$, where ε and ψ are user specified parameters.

The tracker movement in each timestep is determined by the number of agents pushing in each direction in that timestep. Let $n_j, j \in \{N, E, S, W\}$ be the number of agents pushing in direction j in a given timestep. The distance, d_j, that the tracker moves in direction j is given by $d_j = \frac{n_j}{n} \times \rho$, where ρ is the step ratio. The step ratio specifies the maximum distance that the tracker can move relative to the target in one timestep. Thus, if $\rho = 2.0$, the tracker can move twice as far as the target in one timestep. If $\rho = 0.75$, the tracker can move 75% of the distance that the target can move in one time step.

4 Experimental Details

We compare the performance of the three swarm configurations on four problem scenarios. Each problem scenario is represented as a target path.

- `zigzag`: Target alternates between moving approximately northeast and moving approximately southeast.
- `scurve`: Target moves from west to east in a serpentine pattern.
- `sharp`: Target direction is randomly initialized. In each timestep, target has a 5–10% chance of changing to a random new direction; otherwise, target continues in current direction.

- random: Target direction is randomly initialized. In each timestep, target direction is changed by an angle drawn from a Gaussian distribution.

Table 1. Fixed parameter settings.

Parameter	Value
Population size, n	50
Number of timesteps	500
Maximum threshold range, R	10
Threshold decrease, ε (learning factor)	0.1
Threshold increase, ψ (forgetting factor)	0.033

The zigzag and sharp paths produce significant periods of constant task demands punctuated by occasional abrupt changes. The scurve and random paths produce gradually changing task demands. Because of the randomness in the test problems and system behavior, each experiment is composed of 100 runs. Unless otherwise specified, the results for each experiment are averaged over all 100 runs.

Table 1 gives the parameter settings that remain fixed throughout all experiments reported here. The threshold decrease, ε, and increase, ψ, values are set such that total adjusted threshold is conserved; given four tasks, when an agent's threshold decreases for one task, it increases by one third of that amount for the three other tasks. We examine multiple values of step ratio, from $\rho = 0.75$ to $\rho = 3.0$ in increments of 0.25, to examine the impact of agent availability on system performance.

We evaluate system performance based on three evaluation metrics.

1. Tracker path length: The tracker path length provides a measure of how well the tracker followed the target path. The target path length in all experiments reported here is 500. The optimum value for this measure is 500.
2. Average difference: The average difference is the average of the difference between the target and tracker positions in each timestep of a run. This measures the average deviation of the target and tracker paths over a run. The optimum value for this measure is zero.
3. Number of task switches: The number of task switches is the average number of times that agents change tasks during a run, averaged over all agents in the swarm. A task switch is defined as switching from one task to another as well as switching between idle and acting on a task. One of the expected advantages of dynamic thresholds is that they allow agents to dynamically specialize to one or fewer tasks. Thus, specialization should result in agents focusing on a single or fewer tasks, and reduction in the frequency of task switching. The optimum value for this measure is zero.

4.1 Results

Figure 1 compares the performance of the three swarm systems with respect to tracker path length. The top row of plots give the results for the two regular paths, `zigzag` and `scurve`. The bottom row of plots give the results for the two random or irregular paths, `sharp` and `random`. The x-axis of each plot indicates the step ratio, ρ. The y-axis of each plot indicates tracker path length. The optimum path length is 500, as indicated by the dashed line.

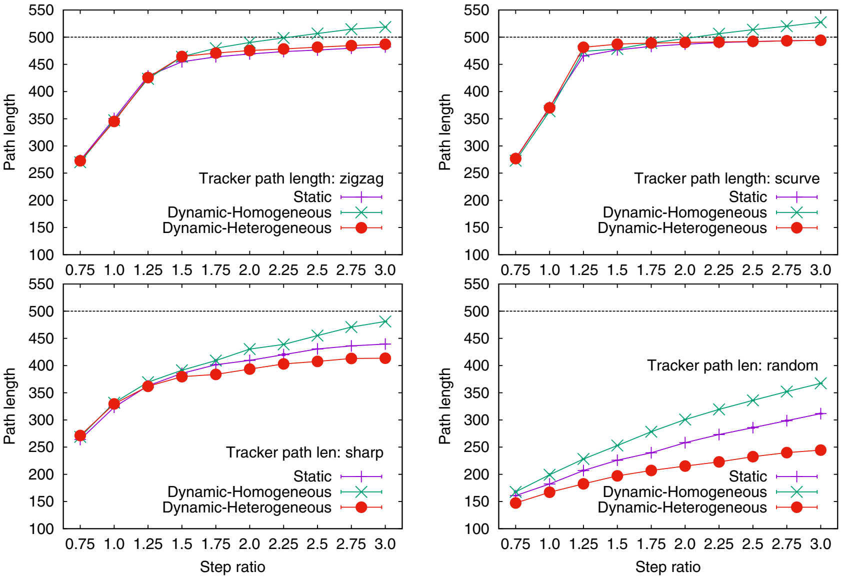

Fig. 1. Average and 95% confidence interval of the tracker path length, averaged over 100 runs. The optimal path length is 500, as indicated by the dashed line. The confidence intervals are extremely tight but they are plotted. (Color figure online)

Comparing the two dynamic systems, the red line and the aqua line, we see that the Dynamic-Homogeneous tracker tends to travel longer paths than the Dynamic-Heterogeneous tracker. As the step ratio increases (as we have more extra agents) this difference increases. On the two regular paths, `zigzag` and `scurve`, Dynamic-Homogeneous overshoots more and more as the step ratio increases. This indicates that more agents than necessary are specializing on tasks and the swarm is likely repeatedly over-shooting and over-correcting the tracker path. Once there are sufficient agents to meet task demands, Dynamic-Heterogeneous and Static both converge gradually toward the optimum path length without over-shooting as extra agent resources increase. On the irregular paths, `sharp` and `random`, Dynamic-Homogeneous generates path lengths closer to the optimum path length than Dynamic-Heterogeneous. Examination of actual paths, however, reveals that both systems generate similar quality solutions. Figure 2 shows example `sharp` runs for step ratio 3.0, the value at which Dynamic-Homogeneous shows the greatest improvement over Dynamic-Heterogeneous. Both systems track the target similarly well and the extra length of the Dynamic-Homogeneous path is actually due to over-correction choppiness.

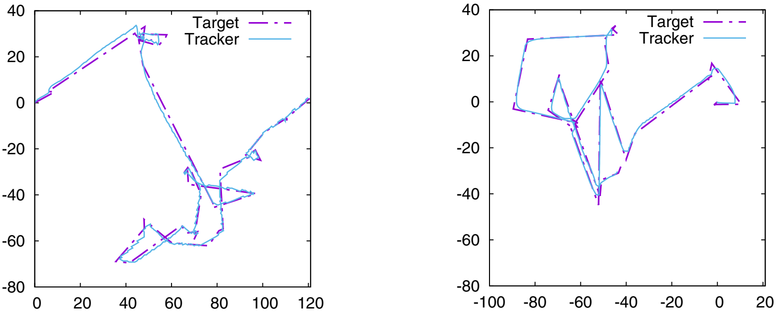

Fig. 2. Target and tracker paths. The left plot is an example Dynamic-Homogeneous run. The right plot is an example Dynamic-Heterogeneous run. Both runs are on the `sharp` path and have a step ratio of 3.0.

Figure 3 compares the performance of the three swarm systems with respect to the average distance between the target and tracker throughout a run. The x-axis of each plot indicates the step ratio, ρ. The y-axis of each plot indicates distance. Comparing the two dynamic systems, we see that when the step ratio is low (there are little or no extra agents), Dynamic-Heterogeneous performs better than Dynamic-Homogeneous, keeping the tracker closer to the target during the run. As step ratio increases (the number of extra agents increase), Dynamic-Homogeneous becomes the better performer. Static continues to perform well relative to the dynamic systems, achieving the best or close to best performance of the three. All three systems performed similarly overall; on a path of length 500 units, all three systems maintained average distances within two units or less of each other for each step ratio value.

Figure 4 compares the performance of the three swarm systems with respect to the average number of task switches per agent per run. The x-axis of each plot indicates the step ratio, ρ. The y-axis of each plot indicates number of switches. In all paths except for `zigzag`, Dynamic-Homogeneous performs significantly worse than either Dynamic-Heterogeneous or Static. In the `zigzag` path, the performance of the two dynamic systems is similar when the step ratio is low, and Dynamic-Heterogeneous becomes significantly better as step ratio increases. Static performs significantly better than either dynamic system on the regular paths. Static's advantage is less consistent on the irregular paths where Dynamic-Heterogeneous outperforms it (undergoes significantly fewer task switches) on the `random` path.

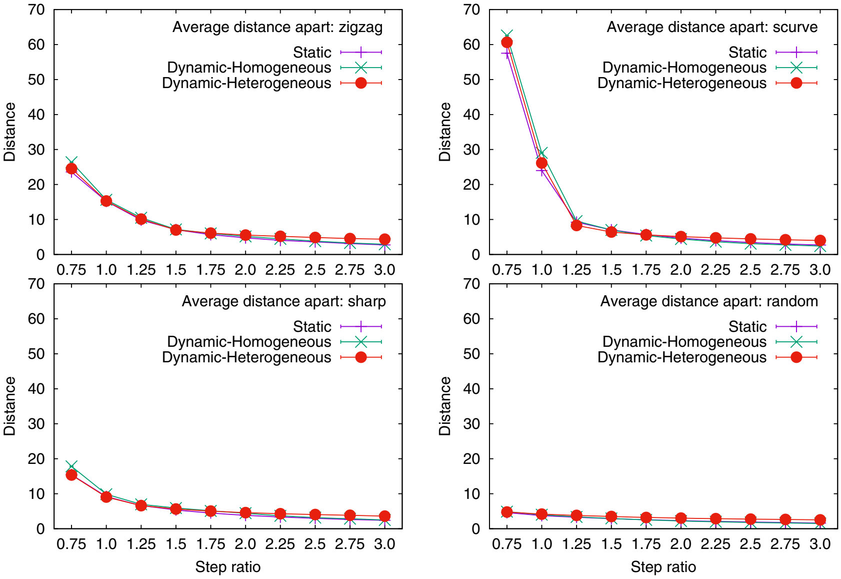

Fig. 3. Average and 95% confidence interval of the average distance between target and tracker during a run, averaged over 100 runs. The confidence intervals are extremely tight but they are plotted.

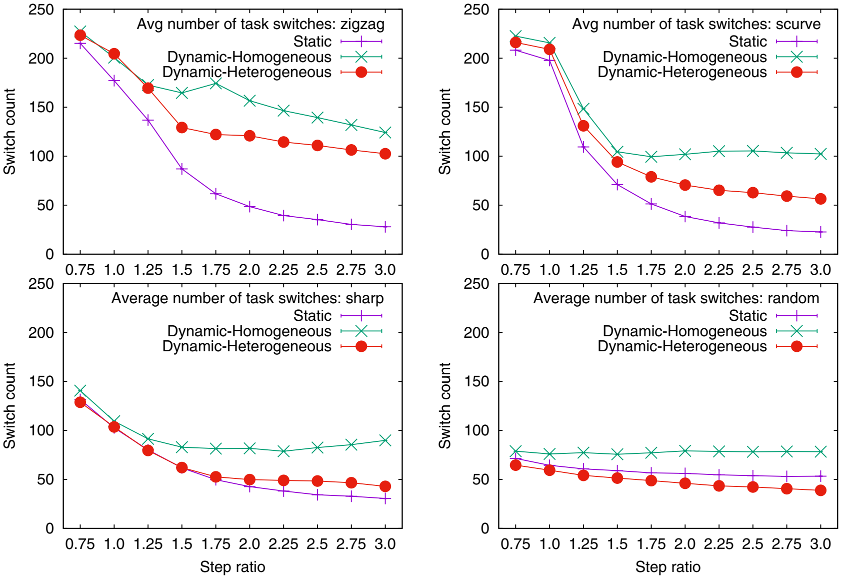

Fig. 4. Average and 95% confidence interval of the average number of task switches per agent during a run, averaged over 100 runs. The confidence intervals are extremely tight but they are plotted.

4.2 Agent Thresholds and Actions

The previous results suggest that Dynamic-Homogeneous and Dynamic-Heterogeneous are able to track the target with similar skill, with Dynamic-Heterogeneous forming more stable specializations. To verify this conclusion, we need to examine how agents act and adapt their thresholds over the course of a run.

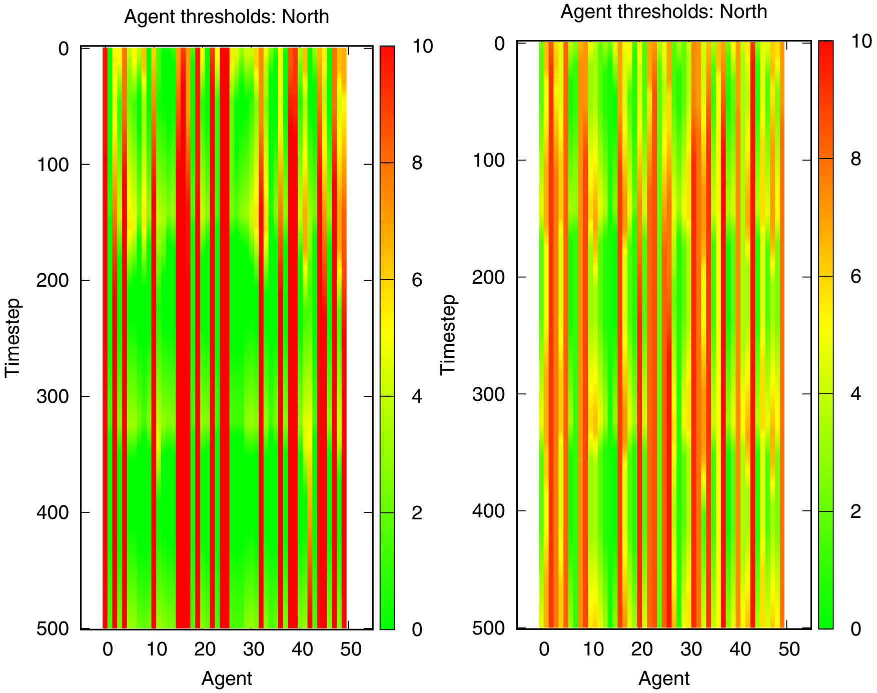

Fig. 5. Threshold values for $\theta_{i,N}$ for all agents over the course of a run. The left plot is an example Dynamic-Homogeneous run. The right plot is an example Dynamic-Heterogeneous run. Both runs are on the `zigzag` path and have a step ratio of 1.5. (Color figure online)

Figure 5 shows how the $\theta_{i,N}$ threshold of all agents in the swarm change over time in two example runs. The left plot is an example Dynamic-Homogeneous run. The right plot is an example Dynamic-Heterogeneous run. Both runs are on the `zigzag` path and have a step ratio of 1.5. The x-axis of both plots indicates agent number, i. The y-axis of both plots indicates timestep. Each column shows the values for one agent's $\theta_{i,N}$ (threshold for pushing north) and how they change over time. Green indicates low threshold (quick to act) and red indicates high threshold (unlikely to act).

Recall that initial thresholds are randomly generated in both systems. Accordingly, there is a mix of colors in the top rows of both plots. As the runs proceed, the Dynamic-Homogeneous agents shown in the left plot clearly converge to extreme threshold values as indicated by the bright red and green values in the second half of the plot. Although early on (top half of the plot), color changes within a column indicates that there are agents that are adapting their thresholds, instances of color changes diminish as the run proceeds and the bottom half of the plot shows much less evidence of threshold adaptation. Once converged to red or green, most agents stay on that color, indicating that their thresholds have become stuck in a sink state.

The Dynamic-Heterogeneous agents shown in the right plot maintain a much more diverse distribution of values throughout the run. Evidence of agents adapting their threshold (color changes within a column) exist throughout the run. When agents converge, the values (colors) to which they converge are less extreme, which allows for a greater possibility of future change. Evidence of the agents reacting to the regular `zigzag` path remains throughout the run in the periodic color shifts.

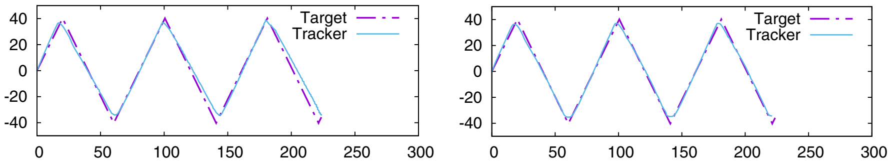

Fig. 6. Target and tracker paths corresponding to the runs from Fig. 5. The left plot is an example Dynamic-Homogeneous run. The right plot is an example Dynamic-Heterogeneous run. Both runs are on the `zigzag` path and have a step ratio of 1.5.

Figure 6 shows the corresponding paths traveled by the target and tracker in the runs from Fig. 5. The left plot is the Dynamic-Homogeneous run. The right plot is the Dynamic-Heterogeneous run. Both runs are on the `zigzag` path and have a step ratio of 1.5. While both systems track the target well, we can see in the left plot that, as Dynamic-Homogeneous agent thresholds converge, the system's tracking ability declines. Notably, Dynamic-Homogeneous continues to track well when travelling northeast, the direction for which its thresholds first begin to adapt. Its ability to track in the southeast direction declines over time, likely due to agent threshold having converged to a distribution optimized for the first set of tasks it encountered. Dynamic-Heterogeneous agents, on the other hand, track the target well throughout the run in both directions while also generating fewer task switches (as indicated in Fig. 4).

5 Conclusions

In this paper, we test the hypothesis that using heterogeneous threshold ranges instead of homogeneous threshold ranges will allow dynamic response threshold swarms to adapt agent thresholds in response to changing task demands while mitigating the problem of convergence to and inability to leave sink states that occurs with homogeneous threshold ranges. We compare the performance of the proposed Dynamic-Heterogeneous approach with two baseline approaches: the existing Dynamic-Homogeneous approach where all agents have the same threshold ranges and the basic Static approach where all agents are assigned uniformly random static thresholds that do not change.

We test these three systems on a collective tracking problem that is modelled after a honeybee thermoregulation task allocation problem. We test four instances of this problem. Two instances generate regular repeated task demands over time. Two instances generate irregular, somewhat random, task demands over time. In each pair of instances, one illustrates periods of stable task demand punctuated by occasional abrupt change, the other illustrates constant gradual change in task demands.

Our results indicate that, in most situations, Dynamic-Heterogeneous performs as well or better than Dynamic-Homogeneous in terms of allocating appropriate numbers of agents to each task demand over time. The Dynamic-Heterogeneous approach results in a significantly more stable swarm in that it significantly reduces the number of times agents switch tasks. This stability is due in part to the fact that the Dynamic-Heterogeneous approach reduces the likelihood of agent thresholds converging and becoming stuck in extreme values or sink states. Avoidance of those sink states allows agents greater ability to re-adapt their thresholds if task demands change. Examination of how agent thresholds adapt over the course of an example run finds that Dynamic-Heterogeneous maintains a more diverse and more adaptable distribution of thresholds than Dynamic-Homogeneous. As seen in previous work, Dynamic-Homogeneous thresholds tend to converge in response to the first set of task demands encountered and have difficulty re-adapting to new task demands. Dynamic-Heterogeneous threshold remain responsive to changes in task demand while converging enough to lower task switching and increase stability.

An interesting and unexpected result that we have not yet discussed is the fact that swarms in which agents are assigned static uniformly distributed thresholds matches or outperforms both dynamic threshold approaches in a large number of the scenarios that we tested. It is this result that prompted us to examine a range of step ratio values. In trying to understand when dynamic thresholds are necessary, we hypothesize that dynamic thresholds are more crucial in systems without extra agent resources. In such systems, an appropriate distribution of thresholds is necessary in order for the swarm to address all task demands in a timely manner. In systems that do have excess agents, inappropriate distributions of thresholds (and agents that stubbornly refuse to leave tasks that do not need attending) have less of an effect because there are plenty of extra agents to take on unaddressed task demands. This hypothesis is borne out in the data from Figs. 1, 3, and 4 that show that Static's performance advantage over Dynamic-Heterogeneous and Dynamic-Homogeneous is always significantly reduced at lower step ratio values where the systems have few to no extra agents.

In summary, our results suggest two general conclusions with respect to swarms that use dynamic response thresholds. First, heterogeneous threshold ranges effectively mitigate the problem of convergence to sink states that occurs with homogeneous threshold ranges, while still retaining the benefits of threshold adaptation. Second, if a swarm is expected to have excess agents, static uniformly distributed thresholds are a simple and effective approach that deserve serious consideration.

Acknowledgements. This work was supported by the National Science Foundation under Grant No. IIS1816777.

References

1. Bonabeau, E., Theraulaz, G., Deneubourg, J.L.: Quantitative study of the fixed threshold model for the regulation of division of labor in insect societies. Proc. Royal Soc. London Biol. Sci. **263**(1376), 1565–1569 (1996)
2. Bonabeau, E., Theraulaz, G., Deneubourg, J.L.: Fixed response thresholds and the regulation of division of labor in insect societies. Bull. Math. Biol. **60**, 753–807 (1998). https://doi.org/10.1006/bulm.1998.0041
3. Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., Dorigo, M.: Self-organized task allocation to sequentially interdependent tasks in swarm robots. Auton. Agent. Multi-Agent Syst. **25**, 101–125 (2014)
4. Brutschy, A., et al.: Costs and benefits of behavioral specialization. Robot. Auton. Syst. **60**, 1408–1420 (2012)
5. Campos, M., Bonabeau, E., Theraulaz, G., Deneubourg, J.: Dynamic scheduling and division of labor in social insects. Adapt. Behav. **8**, 83–96 (2000)
6. Castello, E., et al.: Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. Swarm Intell. **10**, 1–31 (2018). https://doi.org/10.1007/s11721-015-0117-7
7. 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)
8. Cicirello, V.A., Smith, S.F.: Distributed coordination of resources via wasp-like agents. In: Truszkowski, W., Hinchey, M., Rouff, C. (eds.) WRAC 2002. LNCS (LNAI), vol. 2564, pp. 71–80. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45173-0_5
9. 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. Robot. Auton. Syst. **61**, 714–720 (2013)
10. 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)
11. 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)
12. Gautrais, J., Theraulaz, G., Deneubourg, J., Anderson, C.: Emergent polyethism as a consequence of increase colony size in insect societies. J. Theoret. Biol. **215**, 363–373 (2002)

13. Goldingay, H., van Mourik, J.: The effect of load on agent-based algorithms for distributed task allocation. Inf. Sci. **222**, 66–80 (2013)
14. 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)
15. Jones, J.C., Myerscough, M.R., Graham, S., Oldroyd, B.P.: Honey bee nest thermoregulation: diversity promotes stability. Science **305**(5682), 402–404 (2004)
16. Kanakia, A., Touri, B., Correll, N.: Modeling multi-robot task allocation with limited information as global game. Swarm Intell. **10**(2), 147–160 (2016). https://doi.org/10.1007/s11721-016-0123-4
17. Kazakova, V.A., Wu, A.S.: Specialization vs. re-specialization: effects of Hebbian learning in a dynamic environment. In: Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference, pp. 354–359 (2018)
18. Kazakova, V.A., Wu, A.S., Sukthankar, G.R.: Respecializing swarms by forgetting reinforced thresholds. Swarm Intell. **14**(3), 171–204 (2020). https://doi.org/10.1007/s11721-020-00181-3
19. Krieger, M.J.B., Billeter, J.B.: The call of duty: self-organised task allocation in a population of up to twelve mobile robots. Robot. Auton. Syst. **30**, 65–84 (2000)
20. Labella, T.H., Dorigo, M., Deneubourg, J.: Division of labor in a group of robots inspired by ants' foraging behavior. ACM Trans. Auton. Adapt. Syst. **1**(1), 4–25 (2006)
21. Langridge, E.A., Franks, N.R., Sendova-Franks, A.B.: Improvement in collective performance with experience in ants. Behav. Ecol. Sociobiol. **56**, 523–529 (2004). https://doi.org/10.1007/s00265-004-0824-3
22. Lee, W., Kim, D.E.: Local interaction of agents for division of labor in multi-agent systems. In: Tuci, E., Giagkos, A., Wilson, M., Hallam, J. (eds.) SAB 2016. LNCS (LNAI), vol. 9825, pp. 46–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43488-9_5
23. Lee, W., Kim, D.: History-based response threshold model for division of labor in multi-agent systems. Sensors **17**, 1232 (2017)
24. Lerman, K., Jones, C., Galstyan, A., Mataric, M.J.: Analysis of dynamic task allocation in multi-robot systems. Int. J. Robot. Res. **25**, 225–241 (2006)
25. Liu, W., Winfield, A., Sa, J., Chen, J., Dou, L.: Strategies for energy optimisation in a swarm of foraging robots. In: Şahin, E., Spears, W.M., Winfield, A.F.T. (eds.) SR 2006. LNCS, vol. 4433, pp. 14–26. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71541-2_2
26. Liu, W., Winfield, A., Sa, J., Chen, J., Dou, L.: Towards energy optimisation: Emergent task allocation in a swarm of foraging robots. Adapt. Behav. **15**, 289–305 (2007)
27. Meyer, B., Weidenmuller, A., Chen, R., Garcia, J.: Collective homeostasis 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)
28. Price, R., Tiňo, P.: Evaluation of adaptive nature inspired task allocation against alternate decentralised multiagent strategies. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 982–990. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30217-9_99
29. Ravary, F., Lecoutey, E., Kaminski, G., Chaline, N., Jaisson, P.: Individual experience alone can generate lasting division of labor in ants. Curr. Biol. **17**, 1308–1312 (2007)

30. Theraulaz, G., Bonabeau, E., Deneubourg, J.: Response threshold reinforcement and division of labour in insect societies. Proc. Royal Soc. B **265**, 327–332 (1998)
31. Weidenmüller, A.: The control of nest climate in bumblebee (*Bombus terrestris*) colonies: interindividual variability and self reinforcement in fanning response. Behav. Ecol. **15**, 120–128 (2004)
32. Weidenmüller, A., Chen, R., Meyer, B.: Reconsidering response threshold models—short-term response patterns in thermoregulating bumblebees. Behav. Ecol. Sociobiol. **73**(8), 1–13 (2019). https://doi.org/10.1007/s00265-019-2709-5
33. 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: Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (2020)
34. Wu, A.S., Riggs, C.: Inter-agent variation improves dynamic decentralized task allocation. In: Proceedings 31st International Florida Artificial Intelligence Research Society Conference, pp. 366–369 (2018)