Emergent Heterogeneous Strategies from Homogeneous Capabilities in Multi-Agent Systems

Rolando Fernandez, Erin Zaroukian, James D. Humann, Brandon Perelman, Michael R. Dorothy, Sebastian S. Rodriguez, and Derrik E. Asher

1 Introduction

Like human organizations, agent teams are often formed to take advantage of supplementary similarities or complementary differences [7]. Similarity is leveraged by scaling the system size up with homogeneous agents that can work in parallel to increase the task completion rate. Differentiation in heterogeneous systems allows for specialization to complete diverse sub-tasks that can be integrated into completion of the full task. Degree of heterogeneity is a major differentiating factor among multi-agent systems [15]. In addition to heterogeneity based on form factor (e.g., a team of ground and aerial robots) or hardware-defined function [13], there are agent teams with identical hardware but heterogeneous behaviors. For example, it has been shown that a team of 5 robots with identical hardware, whose labor was divided between digging (prying boxes away from the wall) and twisting (clustering boxes in the center of the testbed), was able to cluster groups of boxes more efficiently than homogeneously programmed agent teams [14]. By altering the mix of diggers and twisters, they showed different levels of efficiency and reliability of the resultant systems. Heterogeneous behavior can also be achieved by dynamic state switching, where agents assume different roles based on their local perception of the environment and task needs, even if they are all running the same behavioral algorithms [8]. In the case of human–AI centaur chess teams, amateur human players and their AI teammates are able to achieve better performance than either

R. Fernandez (✉) · E. Zaroukian · J. D. Humann · B. Perelman · M. R. Dorothy · D. E. Asher
US CCDC Army Research Laboratory, Adelphi, MD, USA
e-mail: rolando.fernandez1.civ@mail.mil

S. S. Rodriguez
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

H. R. Arabnia et al. (eds.), *Advances in Artificial Intelligence and Applied Cognitive Computing*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70296-0_37

human grandmasters or supercomputers, by crafting their method of interaction to leverage one another's strengths [6].

Heterogeneity has also been shown to naturally emerge from homogeneity. In some insect species, juvenile members can be deferentially nurtured so that they show marked dimorphism at maturity, enabling differentiation into soldier ants and drones [11]. So we see that heterogeneity can substantially improve system performance and can emerge from various environmental constraints such as nurture or training, hardware, algorithm, or local temporal dependent on-the-fly behavior differentiation.

The source of heterogeneity we study here is trained behavioral heterogeneity from a reinforcement learning (RL) approach. This raises interesting questions. If multiple agents are trained to complete a task over successive trials, do they learn to differentiate their behavior and take advantage of complementary differences? If so, does each agent learn a fixed role, or are the roles distributed dynamically? If a teammate is lost, changed, or compromised, have the other agents learned robust strategies to compensate? Partial answers to these questions can be found in previous results. Changing the reward structure in RL from zero sum to shared rewards can cause qualitatively different behaviors to emerge from the learning agents [16], implying that they are learning to cooperate. Agents may be allowed to train individual heterogeneous algorithms, train as a set with mutually known inputs and actions, or train homogeneously but with differing sensor information so that they make decisions locally. Even when agents are allowed to train heterogeneously, it is difficult to definitively say that they are learning to specialize or even consider their teammates [1]. They may simply be learning to maximize their own reward in a way that generally scales well to group settings (i.e., learning complementary similarities even if supplementary differences are possible) or find strategies that perform well irrespective of their teammates' actions.

In the following sections, the concept of heterogeneous strategies emerging from homogeneous capabilities is demonstrated. In the Methods section, we describe our simulation environment, where we test agents that are guided either by a learning algorithm or fixed strategies. Next, agent performance is shown with probability distributions and statistics for both homogeneous and heterogeneous cases in the Results section. Finally, the Discussion section points to the conclusions that were drawn from the results and provides further avenues of research associated with heterogeneous strategies from homogeneous capabilities in multi-agent systems.

2 Methods

A continuous bounded 2D simulation environment was utilized to train and evaluate a set of four agents (three predators and one prey, represented as circles) per model in the predator–prey pursuit task [4, 10], and a visualization of the task is shown in Fig. 1. The predators scored points (i.e., were given reward during training and evaluated during testing) every time they collided with the prey. Predator agents

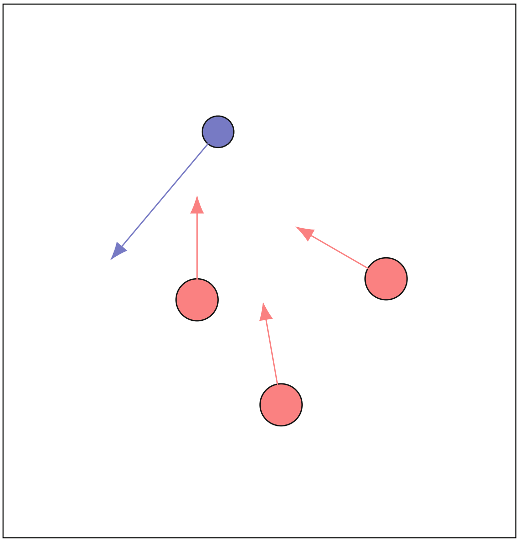


Fig. 1 Predator–prey pursuit particle environment

were homogeneous in their capabilities (i.e., same size, velocity, and acceleration limitations), whereas the prey was 33% smaller, could accelerate 33% faster, and had a 30% max speed advantage. The simulation environment was built upon the OpenAI Gym library [5] and developed for use with the multi-agent deep reinforcement learning algorithm, multi-agent deep deterministic policy gradient (MADDPG) [10]. We assume that the predator agents must cooperate to score a hit on the prey, given the prey's capability advantages. Prior work has shown that a simple greedy policy (i.e., minimize distance to prey) is insufficient for the predators to succeed [3].

All the agents were trained concurrently using the MADDPG algorithm [10]. MADDPG utilizes a decentralized-actor centralized-critic framework that accounts for each agent's observations and actions during training. The predators all received the same fixed reward (shared/joint reward) when any one of them hit the prey, while the prey received the negative of the same fixed reward when hit. At the start of each episode, the initial positions of the agents were randomized, and their initial accelerations and velocities were set to zero. The state space of each predator agent contained its absolute velocity, absolute position in the environment, relative distance and direction to the other predators and the prey, and the prey's absolute velocity. The state space of the prey agent contained its velocity, absolute position in the environment, and relative distance to the predators. The action outputs of the policy network for an agent are accelerations in the two-dimensional coordinate system.

In addition to the MADDPG-trained agents, we consider two analytically defined agents (also referred to in this article as fixed-strategy agents), called a chaser agent and an interceptor agent. The chaser agent does not leverage any prey velocity information and only points its own velocity directly at the prey's instantaneous position. In contrast, the interceptor agent considers both instantaneous position and velocity of the prey. At a moment in time, the Apollonius circle describes the potential interception locations if both agents continue in a constant direction [9]. Given a prey that is faster than the predator, only a subset of possible constant prey strategies admit a capture trajectory for the predator [12]. We extend the Apollonius circle strategy for the predator in the case where capture is not possible. The case where capture is possible is shown in Fig. 2a. Equal travel time at capture gives $\frac{d_E}{V_E} = \frac{d_P}{V_P}$, and the rule of sines gives $\frac{d_P}{\sin\phi} = \frac{d_E}{\sin\theta}$, resulting in

$$\sin\theta = \frac{V_E}{V_P}\sin\phi. \tag{1}$$

In the case where capture is not possible, we consider a finite time prediction for prey trajectory and choose the predator's strategy to minimize the final distance. This is shown in Fig. 2b, where the d_P circle represents how far the pursuer is able to travel in the time it took the evader to travel distance R. The optimum position for the purser to minimize the relative distance at the moment the evader reaches the R circle is to head straight toward that point. Clearly, $\psi = \pi - \theta - \phi$, and

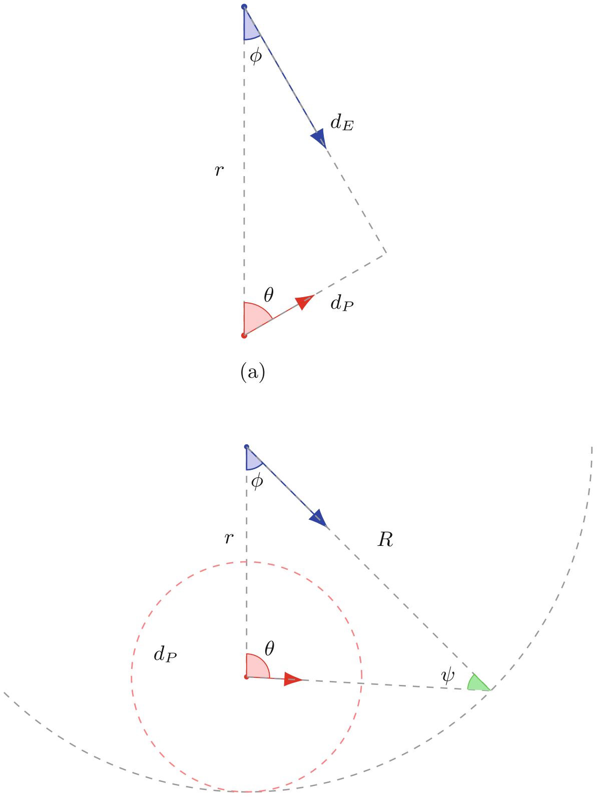
$$\frac{\sin\theta}{R} = \frac{\sin(\theta+\phi)}{r}. \tag{2}$$

The critical case between the capture set and non-capture set is $\sin\phi^* = \frac{V_P}{V_E}$, $\theta^* = \frac{\pi}{2}$, and evaluating Eq. 2 at that point gives

$$R = \frac{r}{\sqrt{1-\frac{V_P^2}{V_E^2}}}. \tag{3}$$

This value for R will result in a continuous policy across all cases.

To compare heterogeneous and homogeneous team structures, the MADDPG-trained agents and fixed-strategy agents were subdivided into homogeneous and heterogeneous teams. We took two previously trained models, independently trained with the MADDPG algorithm for 100,000 episodes and 25 timesteps per episode [2], and used them for our evaluations. Trained Model 1 and 2 predator agents were tested as heterogeneous teams consisting of all three agents (i.e., Agents 0, 1, and 2), labeled as "All Agents" in Figs. 3a, b, or as homogeneous teams in which all three actors' behaviors are driven by the policies of either Agent 0, Agent 1, or Agent 2 from each of the models. Similarly, the fixed-strategy predator agents were tested against both Model 1 and Model 2 prey agents in several types of homogeneous or heterogeneous team compositions: 3 interceptors, 2 interceptors and 1 chaser, 1 interceptor and 2 chasers, and 3 chasers. All tests were performed for 1000 episodes at 1000 timesteps per episode.



(a)

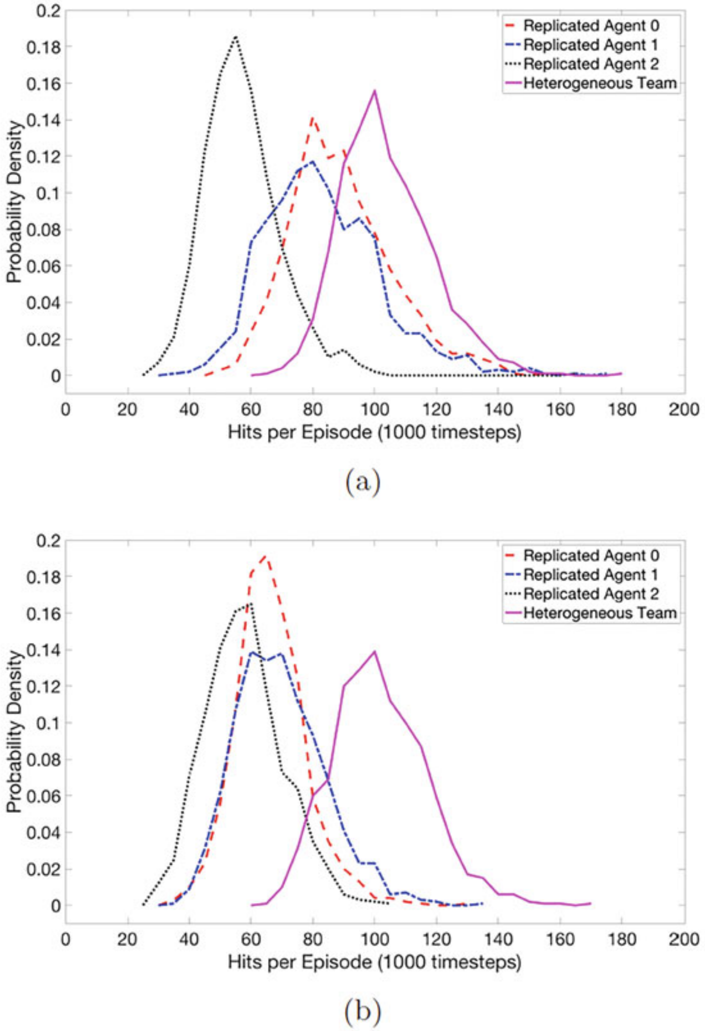
(b)

Fig. 2 Pictorial of interceptor strategy for capture and non-capture cases. (**a**) Capture scenario for interceptor strategy. (**b**) Finite R interceptor strategy

3 Results

The simulation results show how group performance changes upon replicating a single agent's policy network and thus introducing homogeneous strategies. Further, the performance resulting from this homogeneous strategy implementation may provide a means of classifying different trained policies that emerge through collaboration in the multi-agent reinforcement learning process.

Using the data collected during testing, we generated probability density plots of the hits the predators achieved on the prey to analyze agent performance for each of the models (Figs. 3a, b). "All Agents" shows the aggregated or team performance of three predator agents with their independently trained network policies. "Agent 0" represents the data generated from the replication of "Agent 0" across the three predator agents. Similarly, "Agent 1" and "Agent 2," respectively, represent the replication of their corresponding agent policies. The x-axis shows the number of



(a)

(b)

Fig. 3 Probability density performance plots for learning predators. (**a**) Model 1. (**b**) Model 2

hits (i.e., number of times the predators collaboratively contacted the prey agent throughout an episode). The *y*-axis shows the normalized frequency or probability density for the hits per episode. We performed the pairwise 2-sample Kolmogorov–Smirnov (KS) test to show that all the distributions were significantly different from one another at the alpha = 0.01 level (*p*-values « 0.001).

We can see from Figs. 3a, b that in the case of the MADDPG-trained agent strategies the heterogeneous predator team (All Agents) is able to outperform all of the homogeneous predator teams (Agent 0, Agent 1, and Agent 2) where the same policy is replicated across each agent. Furthermore, as all the homogeneous

Table 1 Performance of fixed-strategy predators

Predators' strategy	Mean	CI lower	CI upper
(a) Model 1			
3 interceptors	1.55	1.46	1.65
2 interceptors, 1 chaser	1.16	1.09	1.24
1 interceptors, 2 chaser	0.876	0.824	0.932
3 chasers	0.154	0.145	0.164
(b) Model 2			
3 interceptors	1.44	1.35	1.53
2 interceptors, 1 chaser	1.05	0.990	1.12
1 interceptors, 2 chaser	0.998	0.939	1.06
3 chasers	0.144	0.136	0.153

team performances were significantly different, we can infer that each individual agent policy learned to utilize a different strategy that benefited the team as whole.

The data in Table 1 shows the performance statistics for the fixed-strategy predator agents playing against the MADDPG-trained prey. The statistics were generated by fitting an exponential distribution to the probability density functions of the data from the respective cases. Note that these data are shown in tables rather than plots as the probability density functions are all visually similar when presented with x-axis values on the same scale as Fig. 3a, b.

Overall, the tables allow us to see that the performance of the chaser predators is an order of magnitude worse than that of the interceptor predators when playing against trained prey from Models 1 and 2, which was also shown to be significantly different with 2-sample KS tests (p « 0.001). Interestingly, when we combine a chaser predator with two interceptor predators (i.e., heterogeneous fixed-strategy cases), the inclusion of chaser predators results in a significant reduction in group performance (compare 3 interceptors case to the heterogeneous cases in Table 1). In addition, across both models, combining an interceptor predator with two chaser predators significantly improves group performance from the three chasers case (p « 0.001). Together, these results suggest that overall the interceptor strategy is significantly better than the chaser strategy, especially in the homogeneous cases.

References

1. D. Asher, S. Barton, E. Zaroukian, N. Waytowich, Effect of cooperative team size on coordination in adaptive multi-agent systems, in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006 (International Society for Optics and Photonics, Bellingham, 2019), p. 110060Z
2. D.E. Asher, E. Zaroukian, B. Perelman, J. Perret, R. Fernandez, B. Hoffman, S.S. Rodriguez, Multi-agent collaboration with ergodic spatial distributions, in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, vol. 11413 (International Society for Optics and Photonics, Bellingham, 2020), p. 114131N

3. S.L. Barton, N.R. Waytowich, E. Zaroukian, D.E. Asher, Measuring collaborative emergent behavior in multi-agent reinforcement learning, in *International Conference on Human Systems Engineering and Design: Future Trends and Applications* (Springer, Berlin, 2018), pp. 422–427
4. S.L. Barton, E. Zaroukian, D.E. Asher, N.R. Waytowich, Evaluating the coordination of agents in multi-agent reinforcement learning, in *International Conference on Intelligent Human Systems Integration* (Springer, Berlin, 2019), pp. 765–770
5. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, OpenAI gym (2016, preprint). arXiv:1606.01540
6. N. Case, How to become a centaur. J. Des. Sci. (3). MIT Press. https://doi.org/10.21428/61b2215c (January 8, 2018)
7. H.G. Hicks, C.R. Gullett, S.M. Phillips, W.S. Slaughter, *Organizations: Theory and Behavior* (McGraw-Hill, New York, 1975)
8. J. Humann, Y. Jin, A.M. Madni, Scalability in self-organizing systems: an experimental case study on foraging systems, in *Disciplinary Convergence in Systems Engineering Research* (Springer, Berlin, 2018), pp. 543–557
9. R. Isaacs, *Differential Games* (Wiley, Hoboken, 1965)
10. R. Lowe, Y.I. Wu, A. Tamar, J. Harb, O.P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in *Advances in Neural Information Processing Systems* (2017), pp. 6379–6390
11. G.F. Oster, E.O. Wilson, *Caste and Ecology in the Social Insects* (Princeton University Press, Princeton, 1978)
12. M.V. Ramana, M. Kothari, Pursuit-evasion games of high speed evader. J. Intell. Robot. Syst. **85**(2), 293–306 (2017)
13. D. Shishika, J. Paulos, M.R. Dorothy, M.A. Hsieh, V. Kumar, Team composition for perimeter defense with patrollers and defenders, in *2019 IEEE 58th Conference on Decision and Control (CDC)* (IEEE, Piscataway, 2019), pp. 7325–7332
14. Y. Song, J.H. Kim, D.A. Shell, Self-organized clustering of square objects by multiple robots, in *International Conference on Swarm Intelligence* (Springer, Berlin, 2012), pp. 308–315
15. P. Stone, M. Veloso, Multiagent systems: a survey from a machine learning perspective. Auton. Robot. **8**(3), 345–383 (2000)
16. A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, R. Vicente, Multiagent cooperation and competition with deep reinforcement learning. PloS One **12**(4), e0172395 (2017)