Neuroevolution of Task Specialisation in Multi-Agent Systems

Mostafa Rizk

January 14, 2021

1 Introduction

1.1 Background & Motivation

Multi-agent teams can be used to solve a diverse range of problems such as foraging, collective construction, area coverage and collective manipulation [1, 2, 3]. This research focuses on foraging, when a group of agents must locate and transport resources from one location to another [1, 2, 3], usually a base or nest. Foraging is of particular interest because it has many real-world analogs, especially in robotics, such as search and rescue [1, 2, 3, 4], humanitarian demining [2, 3, 4], hazardous waste cleanup [2] or planetary exploration [2]. Multi-agent foraging is advantageous over single-agent foraging because agents can improve team performance by working in parallel [4, 5] or by cooperating and specialising in different sub-tasks [6, 7, 8], which promises even greater performance.

Learning multiple agent policies that collectively achieve a team objective is challenging, however [9, 10]. Neuroevolution is a promising technique to address this challenge that applies evolutionary algorithms to neural networks to learn policies [11] . Neural networks have shown state of the art performance in several domains [12], including on agent-based problems [13], in part due to their high representational power [14]. Rather than mapping an action to each individual state, like Q-tables, neural networks are able to generalise, making problems with large state spaces much more feasible to solve [14]. Evolutionary algorithms have also had much success, showing comparable or superior performance to state of the art reinforcement learning algorithms [14, 15]. Neuroevolution has the additonal benefit of being highly parallelisable [15] and introduces the possibility of using techniques from the literature such as novelty search [16, 17, 18], architecture search [19, 20] and others to solve more challenging problems [11].

Evolving specialisation has only been done in a few studies [6, 21], though, with most studies opting for less automated approaches [1]. While these studies evolved specialisation successfully, their results are difficult to reproduce and it is not clear what the core challenges of evolving specialisation are. Understanding these challenges can make it easier for future research to produce specialisation, leverage it for foraging and extend it to other multi-agent tasks.

1.2 Research Questions

To better understand how specialisation can be evolved, I investigate the following research question:

How can effective task specialisation be evolved for multi-agent foraging?

To answer this question, I use an experimental framework based on the slope foraging problem used in [6] and similar also to [8, 7]. In this problem, a team of agents must retrieve resources from a source location to a nest. The source is located at the top of a slope and the nest is located at the bottom. Agents can employ a generalist strategy where they go up the slope, pick up the resource, carry it to the nest and deposit it. Agents can alternatively employ a specialist strategy where they cooperate with their team-mates. In the specialist strategy some agents are droppers, who go up to the source and drop resources that fall just short of the nest in an area called the cache, and some agents are collectors, who retrieve resources from the cache and deposit them at the nest. In [8, 7, 6], specialist teams are shown to score higher than generalists because they take advantage of a quick transportation method (i.e. dropping down the slope rather than carrying). I use the slope foraging problem as a test-bed to explore the evolution of specialisation.

I look at three key areas to answer the main research question: the fitness landscape of task specialisation problems, centralised vs decentralised approaches to evolution and theoretical modelling of specialisation. These three areas are explored through the following three sub-questions:

1 What are the fundamental features of the fitness landscape of a task specialisation problem?

Neuroevolution is, fundamentally, an optimisation technique that traverses a landscape and depending on the features of the landscape, different algorithms may be successful. For example, landscapes with a smooth gradient benefit from gradient-based approaches while landscapes with no gradient or multiple optima benefit instead from population-based methods or methods that reward diversity or are more exploratory [22]. Knowing what these features are for a specialisation problem informs what algorithms researchers can use to traverse the landscape and achieve specialisation.

2 When do centralised and decentralised solutions outperform one another in task specialisation problems?

Key to the problem of evolving specialised cooperative teams is the question of centralised vs decentralised evolution [23]. Centralised evolution uses one evolutionary process to evolve a solution for the entire team [23]. Decentralised evolution (cooperative coevolution) uses multiple concurrent evolutionary processes, where each process evolves the solution for an individual agent at the same time as the others are evolving [23]. Centralised vs decentralised evolution is explained in more detail in Section A of the Appendix.

Centralised techniques, theoretically, should be better at finding the best team solution i.e. the team that cooperates most efficiently, but these techniques may not scale with team size because the solution space grows very quickly [23]. Additionally, teams evolved using this approach may not be robust to changes in the individual team members; that is, their cooperative strategy will not work if one of the cooperators is replaced [23].

Decentralised techniques should scale well because each evolutionary process is only looking for a single agent's solution, regardless of the size of the team, so the solution space does not change [23]. These techniques should also be robust because throughout each agent's evolutionary process, it never knows what other agents it will be paired with, so the evolved solution should work well regardless of what the other agents on the team do [23].

Previous works suggests that a decentralised approach is favourable when seeking robustness whereas a centralised one is favourable when seeking cooperation [24], but a thorough analysis has yet to be performed. Moreover, the question of how centralised and decentralised systems compare in general, is still an open one [25]. An empirical comparison of these two approaches in terms of scalability, robustness and degree of specialisation can help determine when it might be preferable to use each approach.

3 What is the price of anarchy in task specialisation problems?

While empirical results are valuable, having a theoretical understanding of them helps to generalise findings. A theoretical model helps interpret results while the results help to refine the model. Evolutionary game theory is commonly used as a modelling tool in the multi-agent domain [23, 24, 26] and provides two useful measures that complement the other two focuses of this research: nash equilibria [27] and the price of anarchy [28]. A Nash equilibrium is a set of strategies, for all players in game, such that no player has an incentive to unilaterally deviate. If the task specialisation problem is formulated as a game, we can expect the Nash equilibria of that game to correlate to the optima in the fitness landscape. This is valuable because the landscape of future specialisation problems can be predicted if they have the same features as a particular game/games. The landscape analysis alone only tells the features of this particular implementation of the task specialisation problem. The landscape analysis paired with the game theoretic model explaining it, tells the features of other instances of the specialisation problem.

The price of anarchy is a measure of how the performance of a socially optimal system, designed by a benevolent dictator, deteriorates when agents can act autonomously to serve their individual interests. The socially optimal system can be considered analogous to a centralised solution and a system of self-interested agents can be considered analogous to a decentralised solution. If the price of anarchy matches what is observed in the empirical setup, it supports the conclusion that these results will follow for other instances of the task specialisation problem. If the price of anarchy does not match, but the model is a correct representation of the problem, then it suggests there are practical issues with implementation that cause a discrepancy.

2 Landscape Analysis

What are the fundamental features of the fitness landscape of a task specialisation problem?

To answer this question, I perform RWG analysis, as proposed in [22], to understand the fitness landscape independently of the architecture used to represent agent policies. This involves randomly sampling several thousand genomes, where each genome represents the neural networks of a team of two agents, evaluating that team over multiple episodes and visualising the distribution of team scores. I can then make inferences about things like the gradient of the landscape and the presence of local optima. Paired with information about each solution's degree of specialisation, I can also make conclusions about how common specialisation is in the landscape and how it correlates to solution quality.

I also use dimensionality reduction to reduce the high-dimensional landscape to a 2D representation that maintains the distances between solutions. This visualisation allows me to make further inferences about the gradient of the landscape and presence of local optima. But it also makes it possible to infer the number of optima and where specialised and/or high-quality solutions are in the landscape relative to non-specialised and/or low-quality solutions.

2.1 Creating a Fitness Gradient

One conclusion of the research so far has been that the landscape of task specialisation problems appears to be very flat and thus difficult to find any solution at all, let alone a specialised one. If a flat fitness landscape is inherent to task specialisation problems, then this is significant because it implies that algorithms that are highly exploratory should be used, as they are with similar problems [22]. However, it is also possible that the fitness function might be making evolution unnecessarily difficult because it only rewards task completion, treating partial success the same as failure. If partial successes are rewarded, the landscape could have a smoother gradient, making it easier for evolution to traverse. This leads to the following hypothesis:

Hypothesis 1.1: Rewarding agents for partial retrieval of the resource will create a smoother fitness gradient.

I conducted an experiment to test this hypothesis where I repeated the landscape analysis but with rewards for partial task completion. The results showed that partial rewards do indeed create a slight gradient but the landscape is still overwhelmingly flat due to a common problem in the literature known as the bootstrapping problem, so the hypothesis is rejected. Section B of the appendix describes the details and results of this experiment and concludes in the discussion that the most appropriate way to address the bootstrapping problem, for this research, is to use the best solution found by RWG as a starting point for evolution, rather than reward partial task completion. This avoids the injection of human bias into the search process and makes the conclusions generalisable to more difficult problem instances.

2.2 Visualising Specialisation

I already performed the initial analysis of the landscape (Figure 1), however, the visualisation did not contain any information about how specialised each team was i.e. was a given solution a pair of generalists or a pair of specialists? Are the better solutions in the landscape usually specialists? To visualise this information I need a way of measuring the degree of specialisation of a solution, so the second task in this group is to evaluate some candidate metrics and the third is to incorporate the chosen metric into the landscape analysis. The hypothesis and experiment are:

Hypothesis 1.2: Specialist solutions outperform generalist ones.

Experiment 1.2- Specialisation Plots of RWG Analysis: Use the chosen specialisation metric to assess all the solutions found by RWG analysis and incorporate them into the plots.

The most specialised solutions are expected to be at the rightmost of the mean plot and above the generalist ones. If this is the case, then it supports the hypothesis. If there are generalist solutions that outperform specialist ones, it disproves the hypothesis.

2.3 Dimensionality Reduction

The neural networks used to solve the Slope Foraging problem have hundreds of weights, meaning that the fitness landscape has just as many dimensions and is consequently very difficult to visualise. Reducing the dimensionality of the problem to two or three dimensions makes visualisation possible and gives many valuable insights into the features of the landscape, especially if combined with a specialisation metric. How far apart are specialist and generalist solutions in the landscape? How many optima are there? What trajectory does evolution take through the landscape? There are many techniques to choose from [29] and further consideration is necessary before choosing one, but Sammon mapping [30] is currently the preferred method due to its ability to preserve the relative distances between solutions and its use in other agent-based and neuroevolution studies [31, 32].
The hypothesis is:

Hypothesis 1.3: There are two optima in the landscape, one composed of specialist solutions and one composed of mostly generalist solutions.

Experiment 1.3- Dimensionality Reduction: Assemble a dataset of solutions, combining some random ones from rwg and some evolved ones. Apply the chosen dimensionality reduction technique to the dataset. Use the chosen specialisation metric to plot the degree of specialisation (in addition to the fitness).

If there are indeed two optima, then the hypothesis is supported, otherwise it is rejected. If, additionally, the specialist solutions are fitter than the generalist ones, Hypothesis 1.2 is further supported.

I have already implemented a version of this and began getting preliminary results.

2.4 Analysis of Different Slope Angles

For the Slope Foraging task, the slope is what provides evolutionary pressure for specialisation to emerge. Having no slope at all means there should be no benefit from specialisation and no pressure for it to evolve. Changing the slope angle changes how beneficial specialisation is. Experiments so far have only focused on one slope setting, so further experiments are needed to examine an environment with no slope and an environment with intermediate slope.
Thus:

Hypothesis 1.4.1: When the slope is flat, there will only be one optima of generalists and no specialist solutions.

Hypothesis 1.4.2: An intermediate slope will have two optima (one specialist and one generalist).

Hypothesis 1.4.3: The specialist solutions will outscore the generalists, but the disparity will be smaller than the steeper slope setting.

Experiment 1.4- RWG with Different Slope Angles: Repeat RWG, as it was done for previous experiments, but with no slope and intermediate slope. Then perform 30 evolutionary runs, each with a different seed genome and random seed. Use the generated data to create RWG analysis plots and perform dimensionality reduction.

If there are any specialist solutions found for the 0 slope setting or more than one optima, Hypothesis 1.4.1 is rejected, otherwise it is supported.

If the intermediate slope setting has more or less than two optima, Hypothesis 1.4.2 is rejected, if it has exactly two, it is supported. If the two optima for the intermediate setting are both generalist, both specialist or both a mix of the two, Hypothesis 1.4.2 is also rejected.

If Hypothesis 1.4.2 is supported and the specialist solutions do not outscore the generalist solutions, Hypothesis 1.4.3 is rejected. If the specialist solutions outscore the generalist solutions but by a margin that is greater than or approximately equal to the margin for the steep slope, Hypothesis 1.4.3 is rejected. If the specialist solutions outscore the generalists by a margin significantly less than the steep slope setting, the hypothesis is supported.

2.5 Decentralised Solutions

So far, RWG analysis has been performed but only for teams produced by a centralised algorithm. It is not possible to perform a meaningful RWG analysis for decentralised algorithms because each solution is a single agent. An individual agent's fitness depends on its team-mates, meaning that the fitness landscape changes and it is not possible to infer its features. What algorithm should be used to navigate the landscape when using a decentralised approach?

One way to gain insight into the fitness landscape for decentralised algorithms is to use dimensionality reduction. Consider 2-agent teams in the centralised setup. A solution in the landscape is the genomes of both agents on the team. Consider now a 2-agent team in the decentralised setup. Each agent was evolved using a separate evolutionary process and navigated its own fitness landscape. However, the end result of the decentralised approach is still a team of two agents, albeit two agents evolved individually. If the genomes of the two agents are concatenated, their team can be plotted in the same landscape as the teams evolved by the centralised approach. This does not tell us what the landscape is like for each agent during the learning process, but it shows us how the decentralised algorithm navigates the shared fitness landscape compared to the centralised algorithm. Does the decentralised algorithm ultimately find the same solutions as the centralised one?

Since centralised learning is trying to find the best team solution and specialisation is what is best for the team, it should more frequently find specialisation than decentralised learning. In decentralised learning, each agent's learning process is trying to find the best solution in an environment where it does not know who it will be partnered with. The best solution should therefore be the one that is most secure for an agent regardless of who their partners are, which is a generalist behaviour. A generalist can get resources on its own and does not need to rely on others.

Hypothesis 1.5: Solutions found by the decentralised algorithm will be primarily generalist whereas solutions found by the centralised algorithm will be primarily specialist.

Experiment 1.5- Decentralised Dimensionality Reduction: Perform 30 evolutionary runs of the decentralised setup. Calculate the degree of specialisation of all evolved solutions. Concatenate the genomes of all individuals on a team and apply dimensionality reduction to the data. Combine the decentralised data with the centralised data and plot on the same plot.

If the majority of solutions found by the decentralised algorithm are not generalists, the hypothesis is rejected. If the majority of solutions found by the centralised algorithm are not specialists, the hypothesis is also rejected. Otherwise, the hypothesis is supported.

3 Centralised vs Decentralised Solutions

When do centralised and decentralised solutions outperform one another in task specialisation problems?

Following from the description of the learning process in Appendix Section A :

3.1 Scalability

Hypothesis 2.1.1: If evolution is run with larger team sizes, the fitness of centralised solutions will decrease.

Hypothesis 2.1.2: If evolution is run with larger team sizes, the fitness of decentralised solutions will decrease less rapidly than centralised solutions

Hypothesis 2.1.3 If evolution is run with larger team sizes, the centralised approach will find fewer specialists.

Hypothesis 2.1.4 If evolution is run with larger team sizes, the decentralised approach will find the same number of specialists.

For centralised learning, since it is searching for a team solution, the search space grows rapidly as the number of agents increases, so we can expect learning to find worse solutions for larger teams. These solutions should be less fit and it can also be expected that it will be more difficult to find specialisation, if those solutions are indeed fitter. Since the decentralised approach is not expected to find specialisation in the first place, that is expected to continue for larger teams.

For decentralised learning, the space of solutions remains the same for each agent, when the team size increases, however, there are more concurrent learning processes. An agent's score depends on many other agents executing unknown strategies in its environment, and all those strategies affect one another, so the increased noise may also make it difficult to find good solutions for larger teams. The time taken by the evolutionary process is not considered in the comparison. However, it is worth noting that the decentralised approach will likely have a slower wall-time if the evolutionary processes are not run in parallel, since evolution must be run once for each agent. Multi-threading should be implemented to avoid this problem otherwise computation time will scale linearly with the number of agents on the team.

Experiment 2.1- Evolution with Larger Teams: Do 30 evolutionary runs (centralised and decentralised) for teams of 4, 6, 8 and 10 agents. Calculate the specialisation of all evolved solutions. Plot the average fitness per agent and team specialisation for teams in each setup.

If the average fitness per agent increases or remains the same for the centralised setup, Hypothesis 2.1.1 is rejected, otherwise it is supported.

If the average fitness per agent of decentralised solutions increases, Hypothesis 2.1.2 is rejected, otherwise it is supported.

If the centralised solutions contain the same number of specialists (or more) for larger teams, Hypothesis 2.1.3 is rejected, otherwise it is supported.

If the decentralised solutions contain significantly more or fewer specialists, Hypothesis 2.1.4 is rejected, otherwise it is supported.

3.2 Robustness

As stated in Hypothesis 1.5, decentralised learning should find the safest strategy for an agent, i.e. a generalist strategy. This strategy is successful regardless of the strategies of its team-mates. If an agent is evaluated with new team-mates it has never been paired with before, it should still perform reasonably well.

Conversely, an agent from the centralised setup is part of a larger solution. It is likely specialised. We can expect that if the agent is paired with new team-mates, the team performance will suffer. The hypotheses are as follows:

Hypothesis 2.2.1: When evolved teams have team-members replaced, the average fitness per agent decreases for centralised teams.

Hypothesis 2.2.2: When evolved teams have team-members replaced, the average fitness per agent decreases for decentralised teams, but by a smaller margin than centralised teams.

Hypothesis 2.2.3: When evolved teams have team-members replaced, the team specialisation decreases for centralised teams.

Hypothesis 2.2.4: When evolved teams have team-members replaced, the team specialisation remains the same for decentralised teams.

Experiment 2.2- Changing Team-mates: Take all 60 evolved 2-agent teams (30 centralised + 30 decentralised). For each evolved team, replace one agent 4 times: once with a dropper, once with a collector, once with a generalist and once with a stationary agent. Calculate the average fitness and team specialisation then repeat for the other agent on the team. That is a total of $2 \times 30 \times 4 \times 2 = 480$ fitness evaluations. Create a violin plot for the centralised setup and one for decentralised setup. Draw one violin for the original team and one for each replacement (a total of five). The data for each violin includes when each of the two agents was replaced. The y-axis shows the fitness and the plots also have a colour scale to show the degree of specialisation.

If the average fitness of centralised solutions increases or remains the same, Hypothesis 2.2.1 is rejected, otherwise it is supported.

If the average fitness of decentralised solutions significantly increases, Hypothesis 2.2.2 is rejected. If the average fitness of decentralised solutions decreases by a larger margin than the centralised solutions, Hypothesis 2.2.2. is rejected. Otherwise it is supported.

If the team specialisation of the centralised solutions significantly increases or stays the same, Hypothesis 2.2.3 is rejected. Otherwise it is supported.

If the team specialisation of the decentralised solutions significantly increases or decreases, Hypothesis 2.2.4 is rejected. Otherwise it is supported.

4 Theoretical Modelling

What is the price of anarchy in task specialisation problems?

Theoretically, centralised learning is equivalent to a benevolent dictator that assigns roles to each agent on the team so that the collective solution is socially optimal. Decentralised learning is equivalent to a group of self-interested individuals where each is trying to find what is best for itself and we expect that is less likely to find a socially optimal solution. Calculating the price of anarchy lets us determine how far the team performance drops when agents are self-interested, i.e. how much performance should drop when using decentralised as opposed to centralised learning.

Hypothesis 3.1.1: The slope foraging problem is a stag hunt with two Nash equilibria, where the payoff-dominant equilibrium is specialist solutions and the risk-dominant is generalist solutions.

Hypothesis 3.1.2: Socially optimal teams are more likely to converge to the payoff-dominant Nash equilibrum while the self-interested teams are more likely to converge to the risk-dominant Nash equilibrium.

Experiment 3.1- Game-theoretic Modelling: Formulate a game that models the experimental setup. Find the nash equilibria and study the learning dynamics.

If the game is indeed a stag hunt, i.e it has two Nash equilibria where the payoff-dominant equilibrium corresponds to agents specialising and the risk-dominant equilibrium corresponds to agents defecting and being generalists, then Hypothesis 3.1.1 is supported. Otherwise it is rejected.

If the socially optimal teams converge to the payoff-dominant equilibrium and the self-interested teams converge to the risk-dominant equilibrium, Hypothesis 3.1.2 is supported. Otherwise it is rejected.

5 Timeline

Each group of tasks in this timeline aligns with one of my research questions, with the exception of two. One group is for this document. The other group is for my literature review. For the literature review, I have created a spreadsheet to systematically keep track of the literature and ensure my research is supported by a firm knowledge of the field. Throughout the next few months, I will add relevant papers to the spreadsheet as I delve deeper into each research question, doing brief first passes of each to get an overview and deeper second and third passes for any that are especially relevant.

Roadmap to Pre-submission

Mon, 12/7/2020

1

TASK	PROGRESS	START	END	DURATION	07-Dec	14-Dec	21-Dec	28-Dec	04-Jan	11-Jan	18-Jan	25-Jan	01-Feb	08-Feb	15-Feb	22-Feb	01-Mar	08-Mar	15-Mar	22-Mar	29-Mar	05-Apr	12-Apr
Plan remaining tasks																							🟩
Define all hypotheses	100%	11-Jan	17-Jan	1w						▨													🟩
Devise experiments for each hypothesis	100%	11-Jan	17-Jan	1w						▨													🟩
Write document outlining plan	100%	7-Dec	17-Jan	6w	▨	▨			▨	▨													🟩
Review literature																							🟩
Ongoing review of literature	10%	7-Dec	11-Apr	18w	▨	▨			■	■	■	■	■	■	■	■	■	■	■	■	■	■	🟩
What are the fundamental features of the fitness landscape of a task specialisation problem?																							🟩
Experiment 1.1- RWG Analsis with Modified Fitness	100%	4-Jan	8-Jan	1w					▨														🟩
Compare specialisation metrics and choose the best one	0%	11-Jan	22-Jan	2w						■	■												🟩
Experiment 1.2- Specialisation Plots of RWG Analysis	0%	25-Jan	29-Jan	1w								■											🟩
Compare dimensionality reduction techniques and choose the best one	0%	25-Jan	29-Jan	1w								■											🟩
Experiment 1.3- Dimensionality Reduction	0%	1-Feb	5-Feb	1w									■										🟩
Experiment 1.4- RWG with Different Slope Angles	0%	8-Feb	14-Feb	1w										■									🟩
Experiment 1.5- Decentralised Dimensionality Reduction	0%	17-Feb	28-Feb	2w											■	■							🟩
When do decentralised solutions outperform centralised solutions in task specialisation problems?																							🟩
Implement code fixes (seeding, cooperative coevolution, multithreading)	80%	7-Dec	14-Feb	10w	▨	▨			▨	▨	▨	▨	■	■									🟩
Experiment 2.1- Evolution with Larger Teams	0%	17-Feb	28-Feb	2w											■	■							🟩
Experiment 2.2- Changing Team-mates	0%	3-Mar	14-Mar	2w													■	■					🟩
What is the price of anarchy in task specialisation problems?																							🟩
Experiment 3.1- Game-theoretic Modelling																							🟩
Formulate a game	0%	15-Feb	5-Mar	3w											■	■	■						🟩
Find nash equilibria	0%	8-Mar	26-Mar	3w														■	■	■			🟩
Study learning dynamics	0%	8-Mar	26-Mar	3w														■	■	■			🟩
Perform synthesis on basis of the data	0%	8-Mar	26-Mar	3w														■	■	■			🟩

References

[1] H. Hamann, "Scenarios of swarm robotics," in *Swarm Robotics: A Formal Approach*, pp. 65–93, Springer, 2018.

[2] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[4] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Cooperative Robots and Sensor Networks 2015*, pp. 31–51, Springer, 2015.

[5] J. Ericksen, M. Moses, and S. Forrest, "Automatically evolving a general controller for robot swarms," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2017.

[6] E. Ferrante, A. E. Turgut, E. Duéñez-Guzmán, M. Dorigo, and T. Wenseleers, "Evolution of self-organized task specialization in robot swarms," *PLoS computational biology*, vol. 11, no. 8, p. e1004273, 2015.

[7] G. Pini, A. Brutschy, M. Frison, A. Roli, M. Dorigo, and M. Birattari, "Task partitioning in swarms of robots: An adaptive method for strategy selection," *Swarm Intelligence*, vol. 5, no. 3-4, pp. 283–304, 2011.

[8] G. Pini, A. Brutschy, G. Francesca, M. Dorigo, and M. Birattari, "Multi-armed bandit formulation of the task partitioning problem in swarm robotics," in *International Conference on Swarm Intelligence*, pp. 109–120, Springer, 2012.

[9] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*, vol. 108. Springer, 2008.

[10] G. Francesca, M. Brambilla, A. Brutschy, V. Trianni, and M. Birattari, "Automode: A novel approach to the automatic design of control software for robot swarms," *Swarm Intelligence*, vol. 8, no. 2, pp. 89–112, 2014.

[11] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 2019.

[12] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[13] G. Marcus, "Innateness, alphazero, and artificial intelligence," *arXiv preprint arXiv:1801.05667*, 2018.

[14] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," *arXiv preprint arXiv:1712.06567*, 2017.

[15] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[16] J. Lehman and K. O. Stanley, "Novelty search and the problem with objectives," in *Genetic programming theory and practice IX*, pp. 37–56, Springer, 2011.

[17] J. Gomes, P. Urbano, and A. L. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intelligence*, vol. 7, no. 2-3, pp. 115–144, 2013.

[18] J. Gomes, P. Mariano, and A. L. Christensen, "Novelty-driven cooperative coevolution," *Evolutionary computation*, vol. 25, no. 2, pp. 275–307, 2017.

[19] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *arXiv preprint arXiv:1808.05377*, 2018.

[20] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392*, 2019.

[21] G. S. Nitschke, M. C. Schut, and A. Eiben, "Evolving behavioral specialization in robot teams to solve a collective construction task," *Swarm and Evolutionary Computation*, vol. 2, pp. 25–38, 2012.

[22] D. Oller, T. Glasmachers, and G. Cuccu, “Analyzing reinforcement learning benchmarks with random weight guessing,” *arXiv preprint arXiv:2004.07707*, 2020.

[23] L. Panait and S. Luke, “Cooperative multi-agent learning: The state of the art,” *Autonomous agents and multi-agent systems*, vol. 11, no. 3, pp. 387–434, 2005.

[24] R. P. Wiegand and M. A. Potter, “Robustness in cooperative coevolution,” in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 369–376, 2006.

[25] R. R. van Lon and T. Holvoet, “When do agents outperform centralized algorithms?,” *Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 6, pp. 1578–1609, 2017.

[26] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel, “A unified game-theoretic approach to multiagent reinforcement learning,” in *Advances in neural information processing systems*, pp. 4190–4203, 2017.

[27] J. Hofbauer and K. Sigmund, “Evolutionary game dynamics,” *Bulletin of the American mathematical society*, vol. 40, no. 4, pp. 479–519, 2003.

[28] E. Koutsoupias and C. Papadimitriou, “Worst-case equilibria,” in *Annual Symposium on Theoretical Aspects of Computer Science*, pp. 404–413, Springer, 1999.

[29] N. Veerapen and G. Ochoa, “Visualising the global structure of search landscapes: genetic improvement as a case study,” *Genetic programming and evolvable machines*, vol. 19, no. 3, pp. 317–349, 2018.

[30] J. W. Sammon, “A nonlinear mapping for data structure analysis,” *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[31] Y.-H. Kim and B.-R. Moon, “New usage of sammon’s mapping for genetic visualization,” in *Genetic and Evolutionary Computation Conference*, pp. 1136–1147, Springer, 2003.

[32] S. Risi, C. E. Hughes, and K. O. Stanley, “Evolving plastic neural networks with novelty search,” *Adaptive Behavior*, vol. 18, no. 6, pp. 470–491, 2010.

[33] F. Silva, M. Duarte, L. Correia, S. M. Oliveira, and A. L. Christensen, “Open issues in evolutionary robotics,” *Evolutionary computation*, vol. 24, no. 2, pp. 205–236, 2016.

[34] Y. Wei, M. Hiraga, K. Ohkura, and Z. Car, “Autonomous task allocation by artificial evolution for robotic swarms in complex tasks,” *Artificial Life and Robotics*, vol. 24, no. 1, pp. 127–134, 2019.

A The Learning Process

Here "learning" refers to evolution.

Each agent is controlled by its own neural network. The network encodes the agent's behaviour i.e. it encodes a function that maps from local observations to actions. The neural network architecture is fixed, but the weights are changed throughout the learning process.

A.1 Centralised

In the centralised setup, a learning algorithm creates a population of solutions. A solution is a sequence of neural network weights for the network of each agent on the team. For example, if a network has 50 weights and there are 4 agents, the solution has a length of 200. Each part of the solution is assigned to a random agent. The agents then carry out several episodes, starting from random positions at the nest each episode and acting for a fixed number of time steps. The score is calculated at the end of each episode and it is the sum of rewards for all resources gathered by the team minus the costs paid by all agents. The average score over all episodes is calculated and used to create a new population of solutions. This repeats until the algorithm terminates.

A.2 Decentralised

The decentralised setup has changed since the seminar and now uses a technique called cooperative coevolution. In cooperative coevolution, rather than one evolutionary process learning a combined solution for the entire team, each agent on the team has its own evolutionary process learning a solution for only itself. For example, if there are 4 agents, there are 4 evolutionary processes running concurrently. At generation i, each of the 4 processes creates a population of solutions for its agent, where a solution is the weights of a single agent's neural network. The score of a solution at generation i is calculated by pairing it with the best solutions from each of the other 3 agents' evolutionary processes at generation $i-1$. The score is for that single agent's performance; it is the team reward (total resources collected) divided by the number of agents (regardless of their contribution) minus the cost incurred by that particular agent.

A.3 Difference

The difference between centralised and decentralised is that a centralised algorithm uses one learning process to learn a solution for the entire team whereas the decentralised algorithm uses multiple concurrent learning processes to learn solutions for each agent independently that are then combined. Centralised learning rewards the team while decentralised learning rewards the individual.

B Creating a Fitness Gradient

Hypothesis 1.1: Rewarding agents for partial retrieval of the resource will create a smoother fitness gradient.

This hypothesis is answered by conducting the following experiment:

Experiment 1.1- RWG Analysis with Modified Fitness: Modify the fitness function such that agents are rewarded for partially retrieving a resource. Specifically, divide the reward for retrieving a resource by the length of the arena and give agents part of the reward if they move the resource a single tile closer to the goal. Then conduct RWG analysis as described in Section 2 and compare the plots to those produced for the original fitness function.

If the mean curve of the distribution has a smoother gradient, then the hypothesis is proven.

The plots for the original landscape analysis and the analysis with the new fitness function are shown in Figure 1 and Figure 2 respectively.

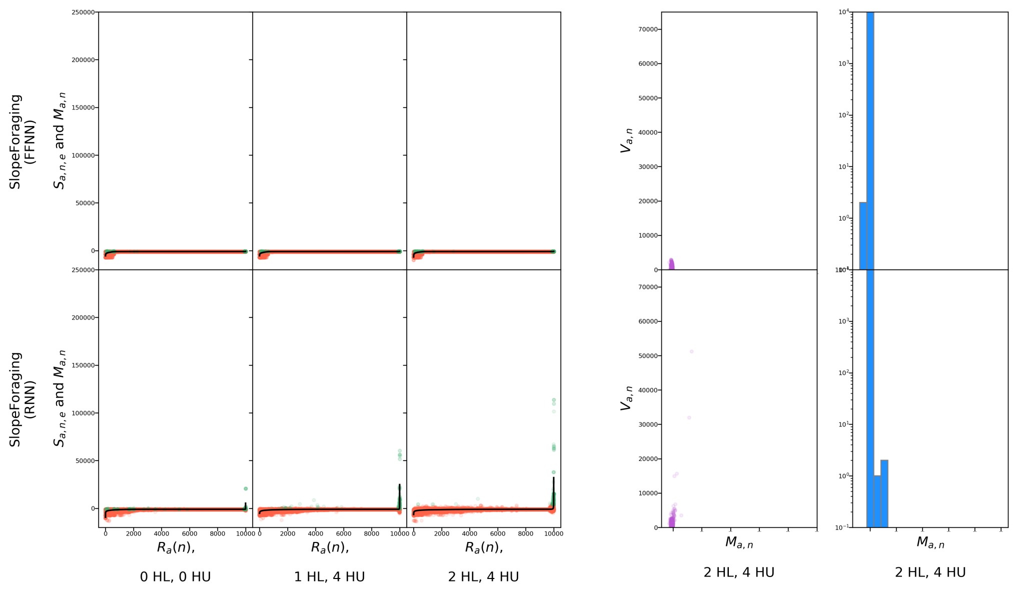

Figure 1: RWG Analysis with the original fitness function

Looking at the 6 mean plots in Figure 2 compared to Figure 1, we see that with incremental rewards, the solutions do in fact form more of a gradient towards the right end of the plot, in particular for RNNs (bottom row). This means that a lot of the solutions in the landscape are partially successful at retrieving resources. However, the landscape is still largely flat. This is presumably because most solutions in the landscape don't even get to the point of picking up a resource from the source.

Thus the answer to Hypothesis 1.1 is:

Answer 1.1: Rewarding agents for partial retrieval of the resource creates a smoother fitness gradient, but the landscape is still largely flat.

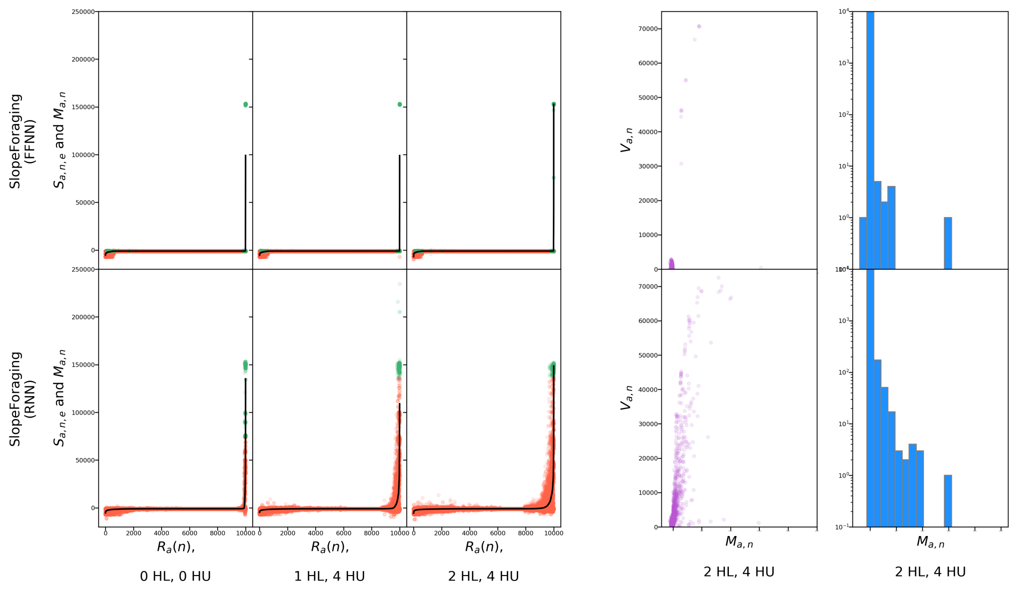

Figure 2: RWG Analysis with incremental rewards

In order to further smooth the fitness landscape there would need to be rewards for partial progress up the slope, but the problem with this is that it is likely to disincentivise the collector behaviour in evolution. Why would an agent stay at the nest and collect resource when they are rewarded for going up the slope? Additionally, while this problem is simple enough to smooth the fitness gradient in this way, such smoothing may not be possible for other more complex problems. The purpose of this research is to understand how to use AI to find good solutions with minimal (if any) human intervention.

Moreover, the flatness of the landscape is, to some extent, inherent to this task. This is a problem common in evolutionary computation known as the bootstrapping problem [33, 34]. Bootstrapping is "when the task is too demanding for the fitness function to apply any meaningful selection pressure on a randomly generated population of initial candidate solutions" [33]. The bootstrapping problem often occurs when the goal behaviour is complex relative to the very simple available actions [34]. For this problem, the available actions are primitives like 'move forward one tile' and 'move backward one tile'. When putting together sequences of these actions, most sequences will obviously not be very successful. There are many proposed solutions to solving the bootstrapping problem and they fall under the broad categories of inserting human knowledge into the learning process or increasing the diversity of solutions [33]. Smoothing the landscape falls under the former category, but this family of techniques has some shortcomings, in particular it reduces the potential for automation and risks the experimenter introducing negative biases. Using rwg to find a seed falls under the second category; it has the shortcoming of additional computation, which may not scale well for problems with larger solution spaces, but for those problems there is a wealth of alternative techniques for increasing diversity, such as novelty search. In keeping with the spirit of AI, I have chosen to continue using the more challenging fitness function.

C List of Hypotheses

1 What are the fundamental features of the fitness landscape of a task specialisation problem?

Hypothesis 1.1: Rewarding agents for partial retrieval of the resource will create a smoother fitness gradient.

Hypothesis 1.2: Specialist solutions outperform generalist ones.

Hypothesis 1.3: There are two optima in the landscape, one composed of specialist solutions and one composed of mostly generalist solutions.

Hypothesis 1.4.1: When the slope is flat, there will only be one optima of generalists and no specialist solutions.

Hypothesis 1.4.2: An intermediate slope will have two optima (one specialist and one generalist).

Hypothesis 1.4.3: The specialist solutions will outscore the generalists, but the disparity will be smaller than the steeper slope setting.

Hypothesis 1.5: Solutions found by the decentralised algorithm will be primarily generalist whereas solutions found by the centralised algorithm will be primarily specialist.

2 When do centralised and decentralised solutions outperform one another in task specialisation problems?

Hypothesis 2.1.1: If evolution is run with larger team sizes, the fitness of centralised solutions will decrease.

Hypothesis 2.1.2: If evolution is run with larger team sizes, the fitness of decentralised solutions will decrease less rapidly than centralised solutions

Hypothesis 2.1.3 If evolution is run with larger team sizes, the centralised approach will find fewer specialists.

Hypothesis 2.1.4 If evolution is run with larger team sizes, the decentralised approach will find the same number of specialists.

Hypothesis 2.2.1: When evolved teams have team-members replaced, the average fitness per agent decreases for centralised teams.

Hypothesis 2.2.2: When evolved teams have team-members replaced, the average fitness per agent decreases for decentralised teams, but by a smaller margin than centralised teams.

Hypothesis 2.2.3: When evolved teams have team-members replaced, the team specialisation decreases for centralised teams.

Hypothesis 2.2.4: When evolved teams have team-members replaced, the team specialisation remains the same for decentralised teams.

3 What is the price of anarchy in task specialisation problems?

> **Hypothesis 3.1.1:** The slope foraging problem is a stag hunt with two Nash equilibria, where the payoff-dominant equilibrium is specialist solutions and the risk-dominant is generalist solutions.
>
> **Hypothesis 3.1.2:** Socially optimal teams are more likely to converge to the payoff-dominant Nash equilibrum while the self-interested teams are more likely to converge to the risk-dominant Nash equilibrium.

D List of Experiments

Experiment 1.1- RWG Analysis with Modified Fitness: Modify the fitness function such that agents are rewarded for partially retrieving a resource. Specifically, divide the reward for retrieving a resource by the length of the arena and give agents part of the reward if they move the resource a single tile closer to the goal. Then conduct RWG analysis as described in Section 2 and compare the plots to those produced for the original fitness function.

Experiment 1.2- Specialisation Plots of RWG Analysis: Use the chosen specialisation metric to assess all the solutions found by RWG analysis and incorporate them into the plots.

Experiment 1.3- Dimensionality Reduction: Assemble a dataset of solutions, combining some random ones from rwg and some evolved ones. Apply the chosen dimensionality reduction technique to the dataset. Use the chosen specialisation metric to plot the degree of specialisation (in addition to the fitness).

Experiment 1.4- RWG with Different Slope Angles: Repeat RWG, as it was done for previous experiments, but with no slope and intermediate slope. Then perform 30 evolutionary runs, each with a different seed genome and random seed. Use the generated data to create RWG analysis plots and perform dimensionality reduction.

Experiment 1.5- Decentralised Dimensionality Reduction: Perform 30 evolutionary runs of the decentralised setup. Calculate the degree of specialisation of all evolved solutions. Concatenate the genomes of all individuals on a team and apply dimensionality reduction to the data. Combine the decentralised data with the centralised data and plot on the same plot.

Experiment 2.1- Evolution with Larger Teams: Do 30 evolutionary runs (centralised and decentralised) for teams of 4, 6, 8 and 10 agents. Calculate the specialisation of all evolved solutions. Plot the average fitness per agent and team specialisation for teams in each setup.

Experiment 2.2- Changing Team-mates: Take all 60 evolved 2-agent teams (30 centralised + 30 decentralised). For each evolved team, replace one agent 4 times: once with a dropper, once with a collector, once with a generalist and once with a stationary agent. Calculate the average fitness and team specialisation then repeat for the other agent on the team. That is a total of $2 \times 30 \times 4 \times 2 = 480$ fitness evaluations. Create a violin plot for the centralised setup and one for decentralised setup. Draw one violin for the original team and one for each replacement (a total of five). The data for each violin includes when each of the two agents was replaced. The y-axis shows the fitness and the plots also have a colour scale to show the degree of specialisation.

Experiment 3.1- Game-theoretic Modelling: Formulate a game that models the experimental setup. Find the nash equilibria and study the learning dynamics.