Diversifying behaviors for learning in Asymmetric Multiagent Systems

Gaurav Dixit
Oregon State University
Corvallis, USA
dixitg@oregonstate.edu

Everardo Gonzalez
Oregon State University
Corvallis, USA
gonzaeve@oregonstate.edu

Kagan Tumer
Oregon State University
Corvallis, USA
kagan.tumer@oregonstate.edu

ABSTRACT

To achieve coordination in multiagent systems such as air traffic control or search and rescue, agents must not only evolve their policies, but also adapt to the behaviors of other agents. However, extending coevolutionary algorithms to complex domains is difficult because agents evolve in the dynamic environment created by the changing policies of other agents. This problem is exacerbated when the teams consist of diverse asymmetric agents (agents with different capabilities and objectives), making it difficult for agents to evolve complementary policies. Quality-Diversity methods solve part of the problem by allowing agents to discover not just optimal, but diverse behaviors, but are computationally intractable in multiagent settings. This paper introduces a multiagent learning framework to allow asymmetric agents to specialize and explore diverse behaviors needed for coordination in a shared environment. The key insight of this work is that a hierarchical decomposition of diversity search, fitness optimization, and team composition modeling allows the fitness on the team-wide objective to direct the diversity search in a dynamic environment. Experimental results in multiagent environments with temporal and spatial coupling requirements demonstrate the diversity of acquired agent synergies in response to a changing environment and team compositions.

CCS CONCEPTS

• **Computing methodologies → Multi-agent systems**; **Cooperation and coordination**.

KEYWORDS

Adaptive Team Balancing, Quality Diversity, Multiagent learning, Evolution

ACM Reference Format:
Gaurav Dixit, Everardo Gonzalez, and Kagan Tumer. 2022. Diversifying behaviors for learning in Asymmetric Multiagent Systems. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528860

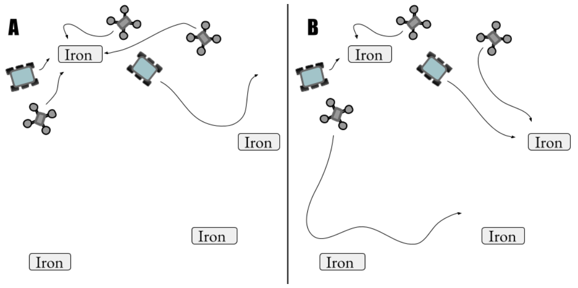

Figure 1: Drone-rover teams collect iron from the Martian surface. Drones can see iron while rovers can collect it with drone supervision. Drones in Team A all exhibit the same behavior: go to the nearest iron. Drones in Team B exhibit diverse behaviors: go to the nearest iron, follow a rover as support, and scout the vicinity. The diversity in B enables faster collection.

1 INTRODUCTION

Many complex real-world problems can be formulated as coevolutionary systems that require agents to evolve diverse policies and work in teams. We've recently witnessed remarkable advances in adaptive agents applied to several multiagent problems such as air traffic control [9, 24], real-time strategy games [17, 25] and robotic automation [11, 13].

In spite of the many successful applications of multiagent learning, the overall generalizability and adoption remains limited. The primary difficulty of most multiagent problems is their asymmetric nature. The asymmetry lies in the distinct capabilities and objectives of different agent types within a team (this is in contrast to heterogeneous agents which may have different capabilities but traditionally share the same objectives). Because no one agent has all the capabilities required to achieve the team-wide objective, the agents must learn to not only complete their objectives but also work together as a cohesive team. This requires agents to learn a diverse set of policies that can be adapted in response to the problem and the capabilities of the other agents. A simple example of this asymmetry is shown in figure 1.

In evolutionary learning, Quality-Diversity methods shift the focus from finding the optimal behaviors to finding diverse behaviors. This shift towards learning diversity is crucial to adaptation in asymmetric multiagent settings. Fundamentally, most Quality-Diversity methods can be described as iterative processes that ping-pong between: 1) Mutating existing behavior(s) to find new diverse behaviors; and 2) Organizing behaviors in an archive. In multiagent settings, exploring the entire behavior space is computationally

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528860

intractable since the behavior space is a function of the number of agents and their action spaces.

Recent work has shown success in scaling Quality-Diversity methods to multiagent settings by using an evolutionary method as a filter to discard parts of the behavior space that have low fitness values on the multiagent task. However, in asymmetric settings, agents can have divergent action spaces implying the need to search through several distinct behavior spaces. Given a fixed budget for team size, this presents the challenge to not only search for diversity but optimize the relative number of different agents in the team.

This work introduces Multiagent Coevolution for Asymmetric Agents (MCAA), a multiagent framework that enables asymmetric agents to coevolve, learn diverse policies, and form robust synergies for cooperation. MCAA combines gradient based Quality-Diversity and explicit team composition optimization with gradient-free evolutionary optimization. The Quality-Diversity method enables agents to discover diverse policies that are trained via a gradient-based optimizer to maximize dense agent-specific rewards. Teams of agents are created by sampling from a distribution over different asymmetric agents, which is updated using a policy-gradient rule to maximize the team-wide fitness. The evolutionary optimizer maximizes the team-wide fitness and acts as a filter for the Quality-Diversity method to discard regions of the behavior space with low fitness. The diversity search and fitness optimization processes operate concurrently and enable the team-wide fitness to guide diversity search.

A key insight is that optimizing for agent diversity rather than performance on a local task finds behaviors well suited for teamwork – that would otherwise be ignored – resulting in higher team performance. This is particularly true for teams of asymmetric agents, which not only have different capabilities like heterogeneous agents, but different objectives as well.

The key benefit of MCAA is its ability to adapt and optimize the team-wide objective by both guiding the diversity search and updating the team composition to reflect the contribution of the discovered diversity to the team-wide objective. We demonstrate the strength of MCAA on an asymmetric multiagent exploration task that requires diverse temporally and spatially coupled agents to work together.

2 BACKGROUND

This section provides a brief overview of recent work in multiagent learning, diversity search methods and open-ended learning.

2.1 Multiagent Learning

One of the key challenges in multiagent learning is solving the credit assignment problem, where the agents need to determine how their policies contributed to the team objective. This problem is more pronounced in tightly coupled domains where the team objective depends on specific joint-actions or in problems with sparse feedback, where a vast majority of policies result in little to no feedback from the environment.

Hierarchical approaches to multiagent learning such as MERL [10] and hierarchical MARL [23] split up learning between agents learning specific behaviors, and learning how to form high performing teams of agents. Tight coupling has been tackled through fitness critics [21], reward shaping [14, 20], and other methods based on an actor-critic architecture and difference rewards [4, 7]. However, none of these methods consider the necessity of diverse agent behaviors in tackling tightly coupled objectives.

2.2 Quality Diversity

Quality Diversity methods guarantee the generation of a diverse set of phenotypes by projecting them into a behavior space to measure the spread of behaviors [2, 5, 6, 16]. If two phenotypes are close in the behavior space, it indicates that they behave similarly, regardless of how different their genotypes may be. This behavior space reduces phenotypes to low dimensional representations that best capture their behaviors, and the features for this behavior space can be engineered or learned.

QD methods such as Novelty Search [8] and MAP-Elites [15] successfully leverage behavior characterizations to generate a diverse repertoire of phenotypes. The MAP-Elites algorithm iteratively mutates genotypes to generate new phenotypes. If a new phenotype demonstrates a higher fitness than another phenotype nearby in the behavior space, the new phenotype is saved and the old phenotype is forgotten. This guarantees that MAP-Elites constantly expands its repertoire of phenotypes to illuminate the behavior space and only holds on to the best phenotype in a given region of that behavior space.

2.3 Open Ended Learning and Coevolution

Open-ended learning continuously modifies a problem such that new solutions must be generated to solve the modified problem. Recent work in open-ended learning has demonstrated the benefits of employing coevolution as a method to simultaneously evolve problems alongside solutions. The Paired Open-Ended Trailblazer (POET) [26] is an approach that combines ideas from behavior novelty and coevolution to build an evolutionary framework for open ended learning [19]. The open ended nature of learning comes not just from the agents, but the problems that evolve with them.

Minimal criterion coevolution (MCC) is another method that is motivated by open-ended coevolution. Instead of defining and comparing fitness of the members of a population, MCC subjects the population to a minimal criterion for reproduction [1]. Requiring a minimum criterion not only eliminates behaviors with lower fitness but the limitation for satisfying an MC also encourages diversity among individual agents. MCC has also been extended to an open-ended search for diversity through speciation that comes naturally as a result of resource limitation [3].

3 METHOD

Multiagent Coevolution for Asymmetric Agents (MCAA) is a coevolution framework for training asymmetric agents to coordinate on a tightly coupled task. Additionally, MCAA functions as an open-ended learning algorithm that adapts agents to changes in the environment. An overview of the framework is shown in figure 2. MCAA produces a set of islands, each with a symmetric agent population for a particular species, and a mainland where agents from each island participate in asymmetric teams. By leveraging both agent-specific feedback on islands and team-wide fitness on the

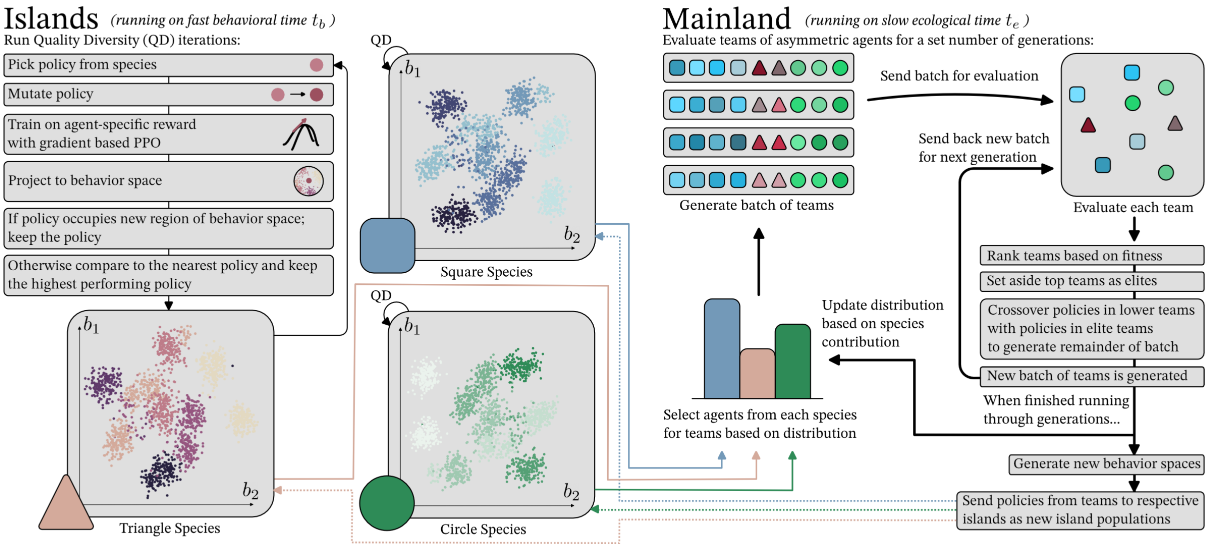
Islands (running on fast behavioral time t_b)
Run Quality Diversity (QD) iterations:
Pick policy from species
Mutate policy
Train on agent-specific reward with gradient based PPO
Project to behavior space
If policy occupies new region of behavior space; keep the policy
Otherwise compare to the nearest policy and keep the highest performing policy
QD
b_1
b_2
Square Species
QD
b_1
b_2
Triangle Species
b_1
b_2
Circle Species
Mainland (running on slow ecological time t_e)
Evaluate teams of asymmetric agents for a set number of generations:
Send batch for evaluation
Send back new batch for next generation
Generate batch of teams
Evaluate each team
Rank teams based on fitness
Set aside top teams as elites
Crossover policies in lower teams with policies in elite teams to generate remainder of batch
New batch of teams is generated
When finished running through generations...
Update distribution based on species contribution
Select agents from each species for teams based on distribution
Generate new behavior spaces
Send policies from teams to respective islands as new island populations

Figure 2: MCAA Overview. On each Island, Quality Diversity iterations produce a population of agents with diverse behaviors. Agents from each island are selected for asymmetric teams on the Mainland according to a species-specific distribution. Generations of team evolution produce agents that contribute to team fitness. The distribution is updated according to the contribution of each species, and agents are sent back to their respective islands for further Quality Diversity. By diversifying species across their underlying behaviors, this approach uncovers otherwise ignored policies that synergistically contribute to high team performance.

mainland, MCAA produces asymmetric agents that specialize their behavior to achieve high team-wide fitness in multiagent teams.

On each island, an agent population of a particular species is trained using a gradient-based optimizer that maximizes an agent-specific reward. A Quality-Diversity routine uses the optimizer to generate a population of high performing agents with a diverse set of behaviors.

On the mainland, a gradient-based softmax distribution determines how many agents from each species are deployed in each asymmetric team. The agents from each species are then chosen randomly from their respective islands. A gradient-free evolutionary algorithm maximizes the team fitness through neuroevolution. This makes it possible to learn robust teams of diverse agents with asymmetric capabilities without having to resort to reward-shaping or explicitly designed team dynamics. The separation of agent-specific diversity and performance optimization on the islands and team dynamics optimization on the mainland also makes it possible for agents with low agent-specific performance but potentially high performance on a team to evolve and learn.

The framework operates on two timescales: a slow ecological time for the evolutionary optimizer on the mainland and a fast behavioral time for the Quality-Diversity optimizer on the individual islands. This gives agents more time to develop basic skills before being evaluated in a team.

Diversity Search On the Islands: An agent species A_i is a population of symmetric agents capable of taking the same actions. An island I_i is inhabited by the corresponding species A_i. Each island starts with a population of P randomly initialized neutral networks as agent policies.

Algorithm 1: Diversity Search

```
Function policy_qd($pop_\pi$):
    Input: population of policies, $pop_\pi$
    Result: population of diverse policies and
            corresponding trajectories, $pop_\pi$ and $pop_{tr}$
    for $qd\_iteration \leftarrow 0$ to $J$ do
        $\pi$ = selection($pop_\pi$) // random policy from population
        $\pi\prime$ = mutate($\pi$)
        $\pi\prime, tr\prime \leftarrow$ train($r_{agent}$) // train with gradient method
          on agent-specific reward, save trajectory
        $bc\prime$ = get_behavior_characterization($tr\prime$)
        $bc \leftarrow$ closest_policy_in_archive()
        /* add_policy, remove_policy modify $pop_\pi$, $pop_{tr}$ */
        if $distance\ (bc\prime, bc) > \lambda$ then
            /* new policy is in a previously unoccupied region
               of behavior space */
            add_policy ($\pi\prime, bc\prime, tr\prime$)
        else
            /* new policy competes locally with the closest
               policy in the behavior space */
            $\pi_c$ = policy($bc\prime$)
            if $reward(\pi_c) < reward(\pi\prime)$ then
                add_policy ($\pi\prime, bc\prime, tr\prime$)
                remove_policy ($\pi_c$)
    return $pop_\pi$,$pop_{tr}$
```

Island	
I	Set of islands
A	Set of agent species
t_b	Behavior time scale index
i	Index of island and corresponding agent species
P	Population size of an island
$k_{(i,j)}$	Individual j of agent species i
Mainland	
T	Set of teams on the mainland
t_e	Ecological time scale index
n	Team index
$k_{(n,m)}$	Individual m of team n
ϕ_k	Fitness of individual k
$\phi_{(i,n)}$	Cumulative fitness of individuals of species i in team n
Distribution Parameters	
μ	Distribution of species on the mainland
w	Parameters of distribution μ
α	Population distribution μ adaptation rate
v	Entropy regularization factor

The policies are trained using a gradient-based optimizer, Proximal Policy Optimization (PPO), on an agent-specific reward [22]. The trajectories of the trained policies (any data associated with the policy can be used) are used as the dataset to train a Dimensionality Reduction (DR) algorithm. This reduced representation of the trajectories acts as a latent characterization of the behaviors and provides the behavior space for the trained policies.

This is followed by many Quality-Diversity iterations:

(1) Select a random policy from the population
(2) Mutate it by probabilistically perturbing weights of the policy network
(3) Retrain the policy on the agent-specific reward
(4) Feed the policy trajectory to the Dimensionality Reduction algorithm to project it in the behavior space
(5) If there are no other policies nearby in the behavior space, then add the policy to the population
(6) If there are policies nearby, then compare the fitness of the new policy to the nearest policy in the behavior space; keep the policy with the higher fitness in the population and exclude the other

The training of policies using the gradient-based optimizer happens on a fast behavioral time scale on each island independently. The ecological time scale determines when agents are pulled from the islands to the mainland, and the process on the mainland sends back a new population of policies with a new behavior space that are fed into more iterations of Quality-Diversity.

Team-Fitness Optimization on the Mainland: On the mainland, at every ecological timestep t_e, $|T|$ teams are created by sampling policies from the islands using the distribution μ. A team is a set of individual policies that are grouped together for evaluation on the team objective. Thus, an individual $k_{(i,j)}$ of species A_i can be a part of several different teams, and its policy network and experience buffer will be the same across the teams. All sampled teams are evaluated on the team-wide fitness on the mainland and each team is assigned a fitness value. A portion of the highest fitness teams is kept aside as elites and the weights of the policies from the remaining teams undergo genetic crossover with the elites through a mutation operator. After a set number of ecological timesteps, the policies from the highest fitness teams along with some crossover policies from lower fitness teams are collected into species populations and sent back to the islands.

Algorithm 2: Multiagent Coordination

```
Function train_teams($K$:Integer, $N$:Integer):
    Initialize $I$ islands, each with a population $pop_{\pi,i}$ of $N$ policy networks
    for $iteration \leftarrow 0$ to $\infty$ do
        /* Fill behavior spaces on all islands using QD */
        for $i \leftarrow 0$ to $I$ do
            $pop_{\pi,i}$, data $\leftarrow$ policy_qd ($pop_{\pi,i}$)
            /* train on local island reward at behavior time scale $t_b$ */
            train_dr (data)
            project_policies()
        Sample from $\sim \mu_{iteration}$ $TxN$ times to create $T$ teams of $N$ agents each
        /* Evolve teams on Mainland at ecological time scale $t_e$ */
        for $generation \leftarrow 0$ to $G$ do
            foreach $team\ n \in T$ do
                $\phi_n$, $\phi_{(I,n)}$ = evaluate ($n$)
            Rank team population $T$ based on fitness $\phi_{n..T}$
            Select the first $e$ teams $\in T$ as elites
            Select the remaining $(M - e)$ teams from $T$, to form set $S$ using tournament selection
            while $|S| < (M - e)$ do
                crossover between randomly sampled policy $\pi_x \in e$ and $\pi_y \in S$ and append to S $\iff$ $\{\pi_x \in i, \pi_y \in i\}$
        $T \leftarrow S \cup e$
        /* update distribution parameters $w$ */
        $w_{iteration+1} = w_{iteration} + \alpha \left[\sum_{i \in \{1,\ldots,|I|\}} \nabla_w \mu(i)(\phi_{(i,n=G)} - vlog\mu(i))\right]$
        for $i \leftarrow 0$ to $I$ do
            project_policies()
```

Behavior Refinement On the Islands: Over the course of several ecological time steps, only a portion of the policies from the islands survive. This information is crucial in guiding the diversity search process on the islands. By discarding policies with low fitness values, the evolutionary method on the mainland essentially acts as a filter to discard policies in regions of the behavior space that do not contribute towards the team fitness. This ensures that the diversity search on the islands is strongly corelated to the team-wide fitness of the island population, even if the agent-specific reward used in the Quality-Diversity process is mis-aligned. For each island, the latent representation of the behavior space is updated by retraining the dimensionality reduction method on the mutated policies. This update ensures that the updated latent representation captures the maximum variance – and thus diversity – of the policies. During the next iteration of Quality-Diversity on the islands, the updated behavior spaces will progressively enable searching for new policies in unexplored but promising regions of the behavior spaces that have been deemed as the most promising on the mainland.

Team Balancing Update: The distribution of species on the mainland, μ, is defined as a softmax over a weight vector ω. At every ecological timestep, t_e, the teams on the mainland are created by sampling individuals from the distribution μ. In an asymmetric multiagent setting, the optimal distribution of species is a function of the current problem and should adapt if the problem changes.

Equations 1 and 2 describe fitness of a species $i = s$ in team n and the average fitness of species $i = s$ across all teams on the mainland at the current ecological timestep t_e, respectively.

$$\phi_{(i=s,n)} = \sum_m (\phi_{k_{(n,m)}} \text{ if } species(k) \text{ is } s) \tag{1}$$

$$f_{(i=s,t_e)} = \frac{\sum_{n \in \{1,\ldots,|T|\}} \phi_{(i=s,n)}}{|T|} \tag{2}$$

The distribution is updated based on the sum of fitness of individuals in a species participating as a team on the mainland, averaged over all sampled teams in the current ecological timestep t_e. The distribution of a species i over the mainland, $\mu(i) = \frac{e^{w_i}}{\sum_l e^{w_l}}$, is updated according to policy gradient rule given by equation 3. The entropy regularization term ensures that each species participates in the mainland problem. This is especially important in the early ecological process as some species might have an overall lower fitness than others.

$$w_{t_e+1} = w_{t_e} + \alpha \left[\sum_{i \in \{1,\ldots,|I|\}} \nabla_w \mu(i)(f_{i,t_e} - vlog\mu(i)) \right] \tag{3}$$

The distribution update takes place at the slower ecological time scale on the mainland. It adapts the distribution of species in sampled teams so that it is best-suited for the current team-wide objective.

4 EXPERIMENTAL SETUP

We evaluate the performance of MCAA on an asymmetric variation of the multiagent rover exploration problem [10]. We conduct the following three experiments to assess the effectiveness of our method on the three problems that it addresses: Diverse behaviors, tightly coupled coordination between asymmetric agents, and adaptability:

1. **Loose Asymmetric Coupling** for tasks that are not tightly coupled but require agents to operate with diverse behaviors, independently.
2. **Tight Asymmetric Coupling** for tasks that require asymmetric agents with diverse policies to coordinate.
3. **Adaptation in Team Composition** to evaluate the composition of teams in response to changes in the environment.

4.1 Asymmetric rover exploration

The classic rover exploration problem consists of symmetric rovers that must explore a continuous two-dimensional space to find and visit points of interest (POIs) that are uniformly distributed in the environment. Every POI has an associated "coupling" constraint which dictates the number of rovers that must visit the POI simultaneously for a successful visit. The optimal strategy in this problem is for the rovers to spread around as teams to visit as many POIs as possible. Real-world multiagent problems are often more diverse in terms of the goals and capabilities required to achieve those goals. We simulate diversity in goals by adding the following POI variants that call for agent specialization and synergies:

1. Vanilla POIs: Standard POIs from the rover exploration task that reward agents with a fixed value (equation 8) when visited with the coupling constraint.
2. Timed POIs: Mission critical POIs that must be prioritized. The reward generated by a Timed POI is reduced by one at every time step.
3. Low-Power POIs: Harder to detect Vanilla or Timed POIs. The probability of a Low-Power POI being encoded in an agent's state space (equation 5) is proportional to the agent's observation radius. Agents with higher observation radius have a higher likelihood of finding these.

Asymmetry in agent capabilities is added by introducing the following agent variants:

Rovers are ground agents that can take three actions: dx, dy (navigational) and obs_r (observation radius to use). The rovers are equipped with two sensors that detect the density of rovers and POIs, given by equations 4 and 5.

$$S_{rover,q} = \sum_{j \in J_q} \frac{1}{d(i,j)} \tag{4}$$

In equation (4), q is the quadrant (the sensor divides the space into 4 quadrants), d measures the Euclidean distance between the sensing rover i and rover j; J_q is the set of all rovers in quadrant q, that are within the observation radius of the rover i.

$$S_{POI,q} = \sum_{k \in K_q} \frac{v_k}{d(i,k)} \tag{5}$$

In equation (5), d measures the Euclidean distance between the sensing rover i and a POI k; K_q is the set of POIs in the quadrant q, within the observation radius of the sensing rover i. The concatenated vectors of densities (all four quadrants) is the rover's input state.

For each episode, the rover starts with a fixed number of energy units e. The amount of energy required at time t, is described by equation 6.

$$e_t = 0.5o_t + 0.3v_t \tag{6}$$

In equation (6), o_t and v_t are observation radius and velocity of the rover at time t, given by the rover's actions.

A rover that uses higher velocity and radius will require more energy and reduce its time of operation in the environment, thus affecting the number of POIs it can visit. A lower velocity ensures more time in the environment but the rover will be less likely to visit Timed POIs. Similarly, a lower observation radius reduces the likelihood of detecting a Low-Power POI at the cost of being able to operate longer and visit other POIs.

Drones are aerial agents that can take three navigational actions: (dx, dy, dz). As aerial units, they have a significantly higher observation radius and can act as scouts on the team to spot POIs. When a Low-Power POI enters a drone's observation radius, it becomes available for rovers to visit regardless of a rover's observation radius. Rovers using a smaller observation radius have a decreased likelihood of finding Low-Power POIs, so they can form good synergies with drones.

The input state vector for a drone is a low resolution image: a stack of 2d vectors, where each vector is a channel. A total of four channels capture the positions of rovers, POIs, comms stations, and other drones, where each agent type has a dedicated channel. The size of the image is directly proportional to the height of the drone dz.

Like rovers, drones also start with a fixed number of energy units e. The amount of energy required at time t, is described by equation 7.

$$e_t = 0.4dz_t + 0.3v_t \tag{7}$$

As described by equation (7), drones can display a variety of behaviors in terms of agility and coverage (size of input image) to manage their energy constraints.

Comms stations are mobile ground agents like rovers, and they can take three actions: dx, dy (navigational) and 'settle'. While in motion, the comms stations have a small, fixed observation radius of 5 units. When a station uses the settle action, the dx and dy actions become unavailable and its observation radius changes to 15 units.

To successfully visit a POI, rovers must fulfil the coupling constraints of the POI and a comms station must have the POI in its observation radius. Thus, the optimal strategy for the comms stations is to spread around the environment and use the 'settle' action. Unlike rovers and drones, the comms stations do not have explicit energy constraints.

The agent capabilities are complementary, which ensures that forming synergies is necessary to visit all POI variants.

Cumulative Team Fitness in this problem is computed using Equation 8.

$$\phi(z) = \sum_k \frac{\prod_i N_{(i,k)} V_k C_k}{\frac{1}{n} \sum_i d(i,k)} \tag{8}$$

$\phi(z)$ is defined for z, the joint state-action of the rovers, drones and comms. V_k is the value of the POI k. $N_{i,k}$ and C_k are indicator functions that are true if rover i is within the observation radius of POI k and a Comms rover has POI k in its observation radius. n is the number of rovers that satisfy the indicator function $N_{i,k}$. Finally, $d(i,k)$ is the Euclidean distance between rover i and POI k.

As dictated by equation 8, for each POI k, the set of n rovers $(r \in i)$ that visit it successfully along with the comms station that satisfied the indicator function C_k, will get the fitness value:

$$\frac{V_k}{(\sum_{r \in i} d(r,k))/n} \tag{9}$$

If POI k was spotted by drone d, then that drone would also receive the fitness given by equation 9.

4.2 Compared Baselines

The quantitative metric of performance for the conducted experiments is the team fitness (equation 8) because MCAA is a multiagent training framework. We are also interested in looking at the diversity of the agent policies and the team composition in response to changes in the environment.

We compare our method with three baselines, each of which serves as a state-of-the-art for a particular facet of the problem: 1) Multiagent Evolutionary Reinforcement Learning (MERL), a hierarchical learning framework that combines gradient-free and gradient-based evolutionary optimization for learning in tightly-coupled (sparse reward) settings [10]; 2) Malthusian Reinforcement Learning, which allows agents to discover cooperative synergies via optimizing their population dynamics [12]; and 3) The intrinsic curiosity module, which uses the prediction error in estimating an agent's next state as a reward for exploration [18].

Our method aims to combine the strengths of all three baselines: It combines gradient-based and gradient-free optimization to maximize team fitness in tightly coupled tasks (like MERL), promotes behavioral diversity via Quality-Diversity, and enables coordination between asymmetric agents by optimizing the distribution of agents in the team (like Malthusian-RL).

4.3 Experiment Parameters

All the results presented below use an environment world size of 50x50 units and every episode lasts for 60 time steps. All agent policies are represented as neural networks. The state space is

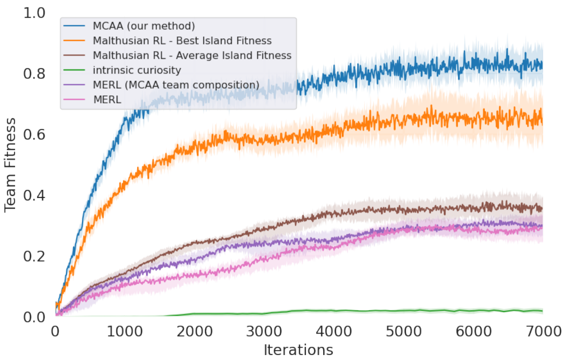

Figure 3: Performance of teams in a loosely-coupled asymmetric task.

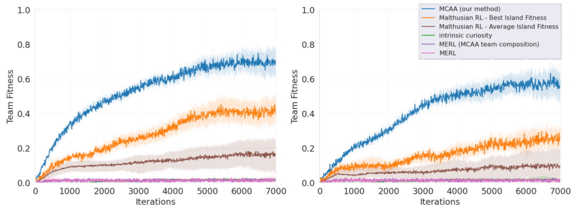

Figure 4: Performance of teams in a tightly-coupled asymmetric task for POI coupling requirement of 3 and 5 respectively.

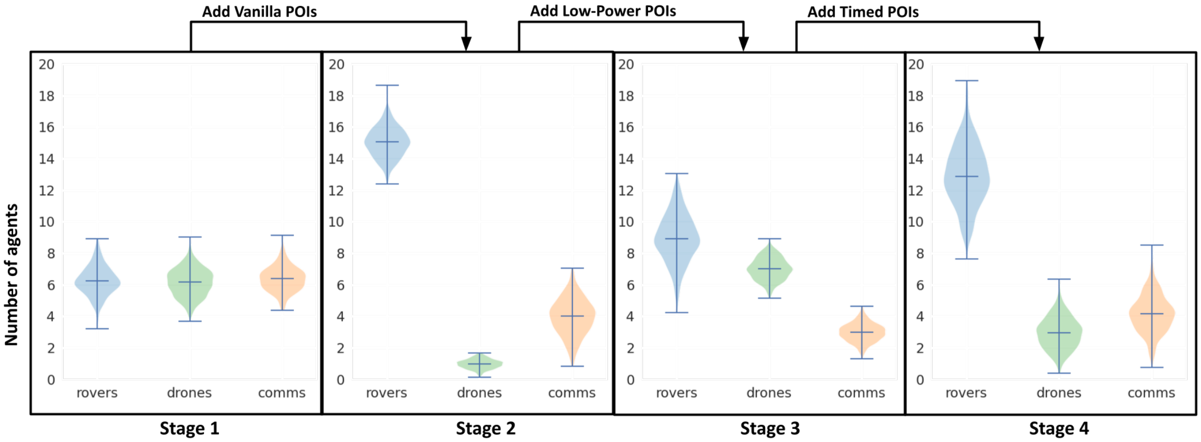

Figure 5: The likelihood of sampling rovers, drones and comms to create teams is shown for three consecutive stages (2000 generations each). Initially, in the first stage, teams have an equal distribution of all three agent types. In the second stage, Vanilla POIs are added which can be visited by the rovers independently. Optimal teams consist of 15 rovers on an average. In the third stage, Low-Power POIs are are added. This shifts the team composition significantly since drones, with a larger observation radius, have a higher probability of finding Low-Power POIs. Finally, Timed POIs are added in stage 4 which causes an increase in both rovers and comms since all three agents must find diverse synergies in order to visit the three POI variants. The change in the team composition and behavioral diversity in response to the changes in the environment highlights the adaptive nature of MCAA.

divided into four quadrants and the **rovers** use an observation space consisting of density estimates of rovers, drones, comms and all POI variants (equations 4 and 5) for each quadrant. The energy e available at each time step is also encoded in the observation vector. The size of the observation vector is $(4 * 3 + 4 * 3 + 1) = 17$. The rover policies map the input state to (dx, dy) actions in $[-2.0, 2.0]^2$ and an observation radius o_t in $[3, 10]$.

The observation vector for the drones consists of a stack of channels, one for each agent type and the amount of energy units available. Each channel is a 2D matrix that one-hot encodes the position of a specific agent type. The size of each channel is proportional to the height of the drone dz. With four channels (one each for rovers, drones, comms and POIs), the size of the observation vector is $(5 * dz)^2 * 4 + 1$. The drone policy networks map the observation vector to (dx, dy) in $[-2.0, 2.0]^2$ and dz in $[1, 3]$.

The observation vector of comms, like the rovers, encode the density of the agents and POI variants. Their policy networks map the observation vector to (dx, dy) in $[-2.0, 2.0]^2$ and $settle$ in $[0, 1]$.

Rovers, drones and comms policies on the island use the dense reward in equation 10 for the Quality-Diversity phase.

$$r_{i,t} = \frac{v_k}{d(i,k)} \tag{10}$$

$d(i, k)$ is the Euclidean distance between the agent i and POI k of value v_k.

In our experiments, Principal Component Analysis (PCA) is used as the dimensionality reduction method. PCA has been used previously to successfully learn the behavior space in single-agent Quality-Diversity methods [6].

For rovers, **The trajectory of the policy** that is used as the input for the dimensionality reduction method is a vector of $(o_t, v_t, d_{r,t}, d_{p,t})$ tuples for every time step t. The observation radius (o_t), current speed (v_t), distance to closest rover $(d_{r,t})$ and the distance to the closest POI $(d_{p,t})$ characterizes the rover's strategy. For drones, the vector at time t consists of current speed (v_t), distance to the closest drone $(d_{d,t})$, distance to the closest POI $(d_{p,t})$ and the height (dz_t). Unlike the rover and drone islands, the island on which comms agents are trained does not use Quality-Diversity, but instead only trains a single policy using the dense reward (equation 10).

5 RESULTS

Several experiments are conducted to evaluate the performance of MCAA in terms of the team fitness and discovered agent behaviors.

5.1 Loose Asymmetric Coupling

In the first experiment, we want to explore the performance of asymmetric agents in teams that do not require tight coordination. This is achieved by setting the coupling constraints of all the POIs to one: a rover can independently visit a POI given a comms agent has the POI in its observation radius.

Figure 3 shows the performance of teams using MCAA and the baselines. Agents are trained with PPO using the distance based dense reward (equation 10) on the islands. On the mainland, the team fitness is given by equation 8. Teams created using MCAA are able to visit about 80% of the POIs. Learning using MCAA is almost completely driven by the learning on the islands that allows rovers and drones to learn diverse policies to go towards POIs. The fitness evaluation on the mainland largely contributes toward optimizing the team composition.

For Malthusian-RL, we use four islands (each hosts the exact same rover environment) with three specie networks (rovers, drones and commms). The dense reward given to agents using Malthusian-RL is generated using the team fitness in equation 9. Teams formed using Malthusian-RL are able to learn as well and visit about 65% of the POIs on the best performing Island. Malthusian-RL struggles to manage the population dynamics since it does not allow the systematic exploration of behaviors like the islands on MCAA. Islands with dominant drones and comms populations were not effective in visiting POIs and significantly reduced the average island performance for this method.

MERL's gradient optimizer uses the dense local reward (equation 10) and its evolutionary optimizer uses the team fitness (equation 8). Rovers that use MERL struggle to learn in this experiment since the method neither encourages agents to find diverse policies nor accounts for agent population dynamics. MERL is also evaluated on teams that are created by sampling teams using the distribution μ learnt by MCAA (figure 3, labeled 'MERL (MCAA team composition)'). This speeds up learning but does not affect its performance significantly.

5.2 Tight Asymmetric Coupling

The next set of experiments evaluate the performance of asymmetric agents in teams on tight-coupled coordination tasks. A higher coupling constraint would require the evolutionary process on the mainland to not only optimize the population distribution but also influence the direction of the Quality-Diversity processes on the islands via biasing the selection of teams towards higher coordinating behaviors.

Figure 4 shows the performance of teams on coupling requirements of five and seven. MCAA uses the same rewards as the previous experiments (equations 8, 10). Teams created using MCAA are able to learn on both tasks although the performance decreases slightly for the higher coupling. Intuitively, this can be explained due to the sparsity of the fitness feedback in the higher coupling task.

Teams using Malthusian-RL struggle to perform well on both these tasks since the ecological time is used exclusively for updating the distribution of species on the mainland. Agents trained with MERL also fail to learn due to the lack of diversity in the agent population.

5.3 Adaptation in Team Composition

Finally, we set up a dynamic environment to test the capacity for MCAA to adapt to the changes in the environment. Adaptation must occur at both the island and mainlands. At the island level, the change in the fitness at a task must initiate diversity search while at the mainland, the population distribution must adapt so as to maximize team-wide fitness on the new task. In our experimental domain, we achieve this by changing the number, location and the coupling requirements of the POI variants in the environment over the course of 4 epochs. The length of each epoch is 4000 ecological time steps (mainland updates), which allowed the population distribution to converge sufficiently in this setup. Figure 5 shows the converged population distributions for rovers, drones and comms for the four epochs.

Before learning starts (epoch = 0), the team composition distribution μ is distributed uniformly. For a fixed team size of 20 agents, initially teams will consist of roughly equal number of rovers, drones and comms agents. This is crucial since it allows the fitness on the team task to make its way to the islands via the behavior refinement step of the method.

In the first epoch (epoch = 1), the environment only consists of Vanilla POIs. The optimal team composition thus only requires rovers (with no particular requirement of diversity in behaviors) to visit the POIs and comms agents to confirm the visits (equation 8). This is reflected in the converged distribution which shows that an average of 16 rovers are present in most teams.

In the second epoch (epoch = 2), the environment consists of Low-Power POIs exclusively, with the coupling constraint set to three. Drones are ideal for observing Low-Power POIs and must be a part of the team. Rovers must learn policies that can operate either with a high observation radius or high velocity in order to be able to directly observe Low-Power POIs or follow drones respectively. The team distribution reflects this with an average of nine rovers, seven drones and three comms sampled per team.

In the final epoch (epoch = 3), the environment has all three POI variants uniformly distributed. The behavior spaces on each island must be sufficiently explored here in order to have diverse policies such as high velocity rovers and high observation radius drones. The converged team sampling distribution indicates that on an average, over half the team consists of rovers (likely for visiting Timed and Vanilla POIs) with a few drones for observing Low-Power POIs.

6 CONCLUSION

We introduced MCAA, a multi-level training framework that enables asymmetric agents in a tightly coupled environment to operate with diverse behaviors and form strong synergies. MCAA combines Quality-Diversity methods with a gradient-free evolutionary optimization. The Quality-Diversity method yields a diverse repertoire of policies that are trained via a gradient-based optimizer to maximize dense agent-specific rewards. An optimal distribution of asymmetric agents required to form high fitness teams is learnt via an evolutionary optimizer that maximizes the team-wide fitness.

The separation of the diversity search and optimization process allows the transformation of team-wide fitness to guide the direction of agent diversity and the optimization of the team sampling distribution.

In this work, MCAA used a fixed allocation and scheduling scheme for computation across the Quality-Diversity, gradient-based and gradient-free optimization phases. In an environment with a stationary team-objective, dynamic resource allocation could potentially speed up the learning process significantly and is a promising step for future work. Finally, we will explore how MCAA can be extended to more complex mixed cooperative-competitive settings by extending the learning scheme to operate on multiple mainlands, each with a different variant of the environment, concurrently.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under grant No. IIS-1815886 and by the Air Force Office of Scientific Research under grant No. FA9550-19-1-0195.

REFERENCES

[1] Jonathan C Brant and Kenneth O Stanley. 2017. Minimal criterion coevolution: a new approach to open-ended search. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 67–74.

[2] Jonathan C. Brant and Kenneth O. Stanley. 2020. Diversity Preservation in Minimal Criterion Coevolution through Resource Limitation. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Cancún, Mexico) *(GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 58–66. https://doi.org/10.1145/3377930.3389809

[3] Jonathan C Brant and Kenneth O Stanley. 2020. Diversity preservation in minimal criterion coevolution through resource limitation. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 58–66.

[4] Jacopo Castellini, Sam Devlin, Frans A Oliehoek, and Rahul Savani. 2020. Difference Rewards Policy Gradients. *arXiv preprint arXiv:2012.11258* (2020).

[5] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. 2020. Scaling MAP-Elites to deep neuroevolution. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Jun 2020). https://doi.org/10.1145/3377930.3390217

[6] Antoine Cully. 2019. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 81–89.

[7] Hoong Chuin Lau Duc Thien Nguyen, Akshat Kumar. 2018. Credit assignment for collective multiagent RL with global rewards. In *Advances in Neural Informaiton Processing Systems (NIPS 2018): Montreal, Canada, December 2-8*. 8102–8113.

[8] Iztok Fister, Andres Iglesias, Akemi Galvez, Javier Del Ser, Eneko Osaba, Iztok Fister, Matjaž Perc, and Mitja Slavinec. 2019. Novelty search for global optimization. *Appl. Math. Comput.* 347 (2019), 865–881. https://doi.org/10.1016/j.amc.2018.11.052

[9] Jared Hill, James Archibald, Wynn Stirling, and Richard Frost. 2005. A multiagent system architecture for distributed air traffic control. In *AIAA guidance, navigation, and control conference and exhibit*. 6049.

[10] Shauharda Khadka, Somdeb Majumdar, Santiago Miret, Stephen McAleer, and Kagan Tumer. 2019. Evolutionary reinforcement learning for sample-efficient multiagent coordination. *arXiv preprint arXiv:1906.07315* (2019).

[11] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

[12] Joel Z. Leibo, Julien Perolat, Edward Hughes, Steven Wheelwright, Adam H. Marblestone, Edgar Duéñez Guzmán, Peter Sunehag, Iain Dunning, and Thore Graepel. 2019. Malthusian Reinforcement Learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1099–1107.

[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[14] Patrick Mannion, Sam Devlin, Jim Duggan, and Enda Howley. 2018. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review* 33 (2018), e23. https://doi.org/10.1017/S0269888918000292

[15] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

[16] Jørgen Nordmoen, Kai Olav Ellefsen, and Kyrre Glette. 2018. *Combining MAP-Elites and Incremental Evolution to Generate Gaits for a Mammalian Quadruped Robot*. 719–733. https://doi.org/10.1007/978-3-319-77538-8_48

[17] OpenAI. 2018. OpenAI Five. https://blog.openai.com/openai-five/.

[18] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR, 2778–2787.

[19] WR Paul, CL William, and KA De Jong. 2002. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. *Journal Spector* (2002), 15.

[20] Aida Rahmattalabi, Jen Jen Chung, Mitchell Colby, and Kagan Tumer. 2016. D++: Structural credit assignment in tightly coupled multiagent domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4424–4429. https://doi.org/10.1109/IROS.2016.7759651

[21] Golden Rockefeller, Patrick Mannion, and Kagan Tumer. 2019. Fitness Critics for Multiagent Learning. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 222–224.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[23] Hongyao Tang, Jianye Hao, Tangjie Lv, Yingfeng Chen, Zongzhang Zhang, Hangtian Jia, Chunxu Ren, Yan Zheng, Zhaopeng Meng, Changjie Fan, and Li Wang. 2019. Hierarchical Deep Multiagent Reinforcement Learning with Temporal Abstraction. arXiv:1809.09332 [cs.LG]

[24] Claire Tomlin, George J Pappas, and Shankar Sastry. 1998. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on automatic control* 43, 4 (1998), 509–521.

[25] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

[26] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. 2019. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753* (2019).