Informed Diversity Search for Learning in Asymmetric Multiagent Systems

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

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

ABSTRACT

To coordinate in multiagent settings, asymmetric agents (agents with distinct objectives and capabilities) must learn diverse behaviors that allow them to maximize their individual and team objectives. Hierarchical learning techniques partially address this by leveraging a combination of Quality-Diversity to learn diverse agent-specific behaviors and evolutionary optimization to maximize team objectives. However, isolating diversity search from team optimization is prone to producing egocentric behaviors that have misaligned objectives. This work introduces Diversity Aligned Island Model (DA-IM), a coevolutionary framework that fluidly adapts diversity search to focus on behaviors that yield high fitness teams. An evolutionary algorithm evolves a population of teams to optimize the team objective. Concurrently, a combination of gradient-based optimizers utilize experiences collected by the teams to reinforce agent-specific behaviors and selectively mutate them based on their fitness on the team objective. Periodically, the mutated policies are added to the evolutionary population to inject diversity and to ensure alignment between the two processes. Empirical evaluations on two asymmetric coordination problems with varying degrees of alignment highlight DA-IM's ability to produce diverse behaviors that outperform existing population-based diversity search methods.

CCS CONCEPTS

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

KEYWORDS

Multiagent Learning, Quality Diversity, Team Composition

ACM Reference Format:
Gaurav Dixit and Kagan Tumer. 2024. Informed Diversity Search for Learning in Asymmetric Multiagent Systems. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3638529.3654206

1 INTRODUCTION

Cooperative multiagent settings are ubiquitous and represent many real-world problems such as healthcare coordination [25], robotic automation [20, 24], and air traffic control [17, 34, 35]. Successful cooperation in these settings requires agents to not only learn good actions, but to learn good joint actions [4]. The problem is aggravated when the agents are asymmetric – they have distinct capabilities and objectives – and must learn diverse inter-agent interactions to overcome potentially conflicting objectives [22, 36].

Quality-Diversity (QD), unlike traditional learning methods, facilitates diversity-seeking optimization by learning and cataloguing a collection of diverse high-performing policies in a behavior space [30]. Coverage, and therefore exploration, is predominantly driven by Genetic Algorithms that repeatedly sample and mutate policies from the behavior space [10]. However, uniformed mutations are inefficient and struggle in high-dimensional behavior spaces [8]. Moreover, exhaustive coverage is intractable and often unnecessary in multiagent settings: only regions of the behavior space that yield policies with the capacity to cooperate are beneficial [11].

Recent advances in multiagent QD methods have employed a hierarchical approach that transforms the behavior space into smaller agent-specific subspaces, making diversity search tractable[11]. However, searching through disjoint agent-specific behavior spaces can produce egocentric behaviors that, although diverse, are misaligned with the team objective.

This work introduces Diversity-Aligned Island Model (DA-IM), a multiagent learning framework that produces teams of asymmetric agents capable of exhibiting diverse high-fitness behaviors required to coordinate. DA-IM leverages a combination of gradient-based optimizers and a gradient-free coevolutionary algorithm which converge simultaneously by sharing information through a behavior space. The coevolutionary algorithm evolves a population of teams (groups of policies) to maximize the team objective. The experiences collected during evolution are used by: 1) A gradient-based off-policy algorithm to sample policy gradients which reinforce behaviors that maximize an agent-specific objective; and 2) An autoencoder that learns a low dimensional representation of the state-action trajectories drawn from the experiences. The resulting latent space is used as a behavior space to perform informed diversity search using an evolution strategy.

Periodically, policies from high-fitness teams and the gradient-based optimizers are added to the shared behavior space. The evolutionary algorithm samples policies from the behavior space therefore allowing diverse policies from the gradient-based optimizers to permeate the team population. Similarly, the experience collected during evolution will feed into the autoencoder to bias diversity search towards regions of the behavior space that yield high-fitness policies.

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0494-9/24/07.
https://doi.org/10.1145/3638529.3654206

We show DA-IM's ability to produce high-fitness teams in two distinct mining problems that require asymmetric agents to balance their individual agent-specific objective with a broader team objective. The ability of our method to produce diverse policies is highlighted in an experiment with misaligned objectives.

2 BACKGROUND AND RELATED WORK

2.1 Multiagent Learning

Learning in multiagent settings is particularly difficult due to non-stationarity and credit assignment [3]: agents in a team must learn simultaneously and discern the impact of their actions on the team reward [21]. The credit assignment problem also extends temporally as multiagent settings are characterized by a sparse reward that is available after taking a long trajectory of actions [37]. Recent advances in reward shaping methods have partially addressed this problem by either creating "stepping stone" rewards or by learning to decompose the sparse reward into denser rewards [27]. However, shaping requires intimate knowledge of the problem and is prone to creating misaligned rewards [13].

The centralized-learning-with-decentralized-execution paradigm addresses this problem partially, by effectively transforming the multiagent system to a single agent Markovian setting [26]. However, this paradigm has seen limited application because of its underlying assumptions of agent symmetry and fixed team size.

Evolutionary methods have been applied to a broad range of multiagent problems because of their ability to optimize a population of policies without using gradients [18]. However, the problem of learning to interact with the environment and with other agents, traditionally occurs along different dimensions (physical and social) [2]. This is exploited by hierarchical methods, such as Multiagent Evolutionary Learning (MERL), that leverage a gradient-based optimizer to learn along the physical dimension and an evolutionary algorithm to learn along the social dimension [19]. MERL implicitly aligns the two objectives by evolving policies that have been trained on agent-specific behaviors. However, this has a homogenizing effect on the population, which is necessary but unsuitable, for the convergence of the two optimizers (explored in section 5.3).

2.2 Quality Diversity

Unlike traditional optimization methods, Quality Diversity (QD) are a family of diversity-seeking methods that catalogue an archive of policies in a behavior space [30]. QD is an iterative two-step process that: 1) samples a policy from the archive and mutates it; and 2) adds the mutated policy to the archive by selecting the higher-fitness policy, if a policy already exists in that space (local K-neighbor distances or niche for structured archives) [8]. It is typically non-trivial to define the behavior space dimensions (archive axes) when the problem exists along the social dimension [2]. Characteristics such as "inclination to cooperate" would be suitable, but are nebulous and difficult to quantify [12].

Recent advances in QD have successfully used dimensionality reduction techniques to infer a behavior space from data collected by a population of policies [10, 12]. However, applying this technique to multiagent settings remains challenging due to the combinatorial nature of multiagent action spaces resulting in a completely impenetrable black-box behavior space [16]. Methods such as Malthusian Reinforcement Learning (MRL) and Multiagent Coevolution for Asymmetric Agents via Island Model (henceforth abbreviated to IM) partially alleviate this by resource limitation, adaptive populations to force specialization, and the transformation of the behavior space into smaller agent-specific spaces [11, 23]. However, these methods are sample inefficient (linearly increasing the number of evaluations with the number of agent classes) and are prone to producing diverse egocentric behaviors that are suboptimal in teams [12]. This work builds off of the Island Model [11] to create a sample efficient learning framework that produces team-aware diversity.

2.3 Evolution Strategies

Evolution Strategies (ES) are gradient-based optimization methods in which a parameterized distribution over solutions is updated in the direction of higher fitness solutions [5]. Our work uses the OpenAI-ES variant that updates its distribution by approximated natural gradients [32]. An isotropic multi-variate Gaussian distribution is used with mean θ_μ and a fixed variance σ. A population of Z solutions is sampled and evaluated on the objective F (algorithm 1, lines 2-3). This is used to estimate the gradient of the expected fitness to update the parameters of the distribution (line 4). We use a novelty score ψ (equation 1) as the objective function. The novelty score for a solution $theta$ is the average distance to its K-nearest neighbors in the behavior space (novelty archive A^N).

$$\psi(\theta) = \frac{1}{K}\sum_{k=1}^{K} ||d_\theta - d_k||_2 \tag{1}$$

Our work uses this ES variant (referred simply as ES henceforth) to systematically search for diverse behaviors in the behavior space.

Algorithm 1: ES Gradient Step (adapted from [32])

1 **Function** `ES_step`(F: *objective*, θ_t: *solution*):
2 $\quad \epsilon_1, ..., \epsilon_Z \sim \mathcal{N}(0, I)$
3 $\quad F_i = F(\theta_t + \sigma\epsilon_i)$, for $i = 1$ to Z
4 $\quad \theta_{t+1} \leftarrow \theta_t + \lambda \frac{1}{Z\sigma}\sum_{i=1}^{Z} F_i\epsilon_i$
5 $\quad$ return θ_{t+1}

3 DIVERSITY ALIGNED ISLAND MODEL

This work introduces Diversity Aligned Island Model (DA-IM), a multiagent learning framework that produces teams of asymmetric agents that express diverse behaviors required to cooperate on individual class-specific and team objectives. This is achieved by leveraging a combination of three processes: 1) an evolutionary algorithm that evolves a population of teams to maximize team fitness, 2) a gradient-based optimization that reinforces experiences generated by the evolutionary algorithm to maximize individual objectives, and 3) an evolution strategy that introduces diversity in the team population by systematically improving coverage in the behavior space. We adopt the terminology introduced in [11]: the evolutionary algorithm used to train teams of asymmetric agents is called a "mainland" and the combination of policy gradients and NS-ES for each agent class is called an "island".

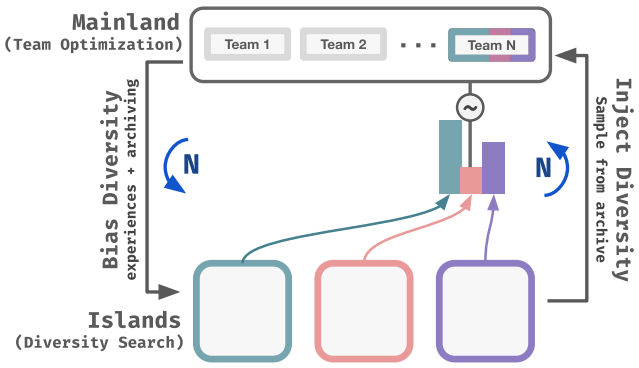

Figure 1: DA-IM Overview: Each island represents a combination of a class-specific objective and a QD optimizer (figure 2). A mainland represents a population of teams (groups of policies) that are evolved to maximize the team objective (fitness $\phi(t)$). After every N gradient updates on the island and evolutionary updates on the mainland, the weights w of the softmax distribution μ are updated in the direction of team composition that maximizes fitness $\phi(t)$ on the mainland. The softmax μ is used to sample policies from the elite archives $E^A_{i\in I}$ from all islands to replace the low-fitness teams on the mainland by potentially diverse policies that were discovered on the islands. Additionally, policies from elite teams on the mainland are added to the elite archives. The experiences collected on the mainland are then migrated to each island to bias the diversity search.

Figure 1 presents a high-level overview of DA-IM. An island i is initialized for each agent class, along with an initial population pop_i of randomly initialized policies (algorithm 2, lines 1,2). Each island $i \in I$ is assigned two empty archives (elite $\mathcal{A}^E_i$ and novelty archive $\mathcal{A}^N_i$), and a replay buffer $\mathcal{R}_i$ that will store experiences collected by the population pop_i (lines 3-4).

DA-IM progresses as follows: Policies from the initial populations pop_i are drawn from a categorical distribution, with probabilities given by a softmax μ over weights w^I, to create M teams of S policies each (line 6). The weights w^I are randomly initialized and dictate the team composition. The mainland runs an evolutionary algorithm over the T initial teams and collects experiences in R_I replay buffers (line 7). The mainland and I islands run in parallel until convergence (lines 8-11).

3.1 Mainland

A mainland maintains a population of teams (groups of policies) that is evolved using an adaptation of the cooperative coevolutionary algorithm (CCEA) [31] (algorithm 3). At each generation, teams are evaluated and ranked using the sparse team fitness (lines 3-4). The e highest-fitness teams are considered elites and are retained for the next generation (line 5). Policies from the remaining teams undergo a single-point crossover with policies from the elites using binary tournament (lines 6-8). The crossover uniformly samples policies from the low-fitness and elite teams to create new teams that have a higher likelihood of succeeding [15]. Policies in the new teams are

Algorithm 2: Diversity Aligned Island Model (DA-IM)

1 Initialize I islands, one island per agent class
2 Initialize I initial populations of policies $pop_{i\in I}$
3 Initialize archives $\mathcal{A}^E_i$ and $\mathcal{A}^N_i$ for each $i \in I$
4 Initialize I empty cyclic replay buffers $\mathcal{R}_{i\in I}$
5 **Function** DAIM(*I:Islands, M:Mainlands*)**:**
6 $T = [T_1, T_2, ..., T_M] \sim Categorical^S_{pop_{i\in I}}(\boldsymbol{\mu})$
7 $T, \mathcal{R}_I = \texttt{mainland}(T, \mathcal{R}_I)$ `// initial experiences`
8 **for** $k \leftarrow 0$ **to** ∞ **do**
9 **do in parallel**
10 $T, \mathcal{R}_I = \texttt{mainland}(T, \mathcal{R}_I) \quad \forall i \in I$
11 $\texttt{island}_i(\mathcal{R}_i, [\mathcal{A}^E_i, \mathcal{A}^N_i]) \quad \forall i \in I$
12 add_to_archives$(T[0:e]_i, [\mathcal{A}^E_i, \mathcal{A}^N_i]) \quad \forall i \in I$
13 $w \leftarrow w + \alpha \left[\sum_{i=1}^{I} \nabla_w \mu(i)(f_i - v \log \mu(i))\right]$
14 $T' = [T'_1, ..., T'_{(|T|-e)}] \sim Categorical^S_{A^E_I}(\boldsymbol{\mu})$
15 $T \leftarrow T[0:e] \cup T'$

mutated by perturbing their weights using Gaussian noise (line 9). Over N generations, the mainland gradually improves the fitness of asymmetric teams on the team objective.

Algorithm 3: Mainland (CCEA, adapted from [19])

Function `mainland`(*T: teams, $\mathcal{R}_I$: Replay Buffers*)**:**
for *generation* $\leftarrow 0$ **to** N **do**
$\Phi_t, \mathcal{R}_I = evaluate(t) \mid \forall t \in T$
$T \leftarrow rank(T)$ `// sort using` $\Phi_{t\in T}$
$E = T[0:e]$ `// select top e elite teams`
Create set $S = \emptyset$ using binary tournament:
while $|S| < (|T| - e)$ **do**
$t \leftarrow crossover(t_x \sim U(E), t_y \sim U(T - E))$
$t \leftarrow mutate(t)$ `// perturb weights`
$S \leftarrow S \cup t$
$T \leftarrow S \cup E$

3.2 Islands

For each agent class, an island uses a combination of QD to explicitly search and catalogue diverse policies [10], and Deep Deterministic Policy Gradient (DDPG) [24] to reinforce primitive behaviors. Figure 2 provides an overview of the island process.

Diversity Search: Instead of discarding the experiences generated by the evolutionary algorithm on the mainland, they are collected in $\mathcal{R}_I$ replay buffers which are assigned to their corresponding islands. Trajectories from the buffer $\mathcal{R}_i$ on each island i, are used as a dataset to train an autoencoder. The dimensionality reduction results in a latent space that captures the variance in the policies that generated the input trajectories [1]. This latent space is used as a behavior archive for performing QD [10, 11]. Each island uses two instances of this space: an elite archive $\mathcal{A}_E$ to catalogue policies from high-fitness teams on the mainland, and a novelty

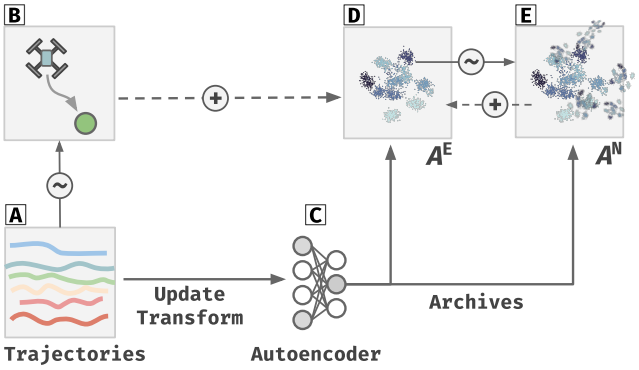

Figure 2: Island Overview: Trajectories of experiences collected on the mainland form a dataset (A). They are sampled by a gradient-based off-policy optimizer to reinforce class-specific objectives (B). An autoencoder is trained on this dataset to produce a low dimensional latent space that is used as the behavior space for QD (A, C). Two distinct instances of this space are used as the elite and novelty archives (D, E). An ES samples from the elite archive and takes N gradient steps to maximize the novelty objective ψ (D, E). Subsequently, Policies from (B) and (E) are added to the elite archive (D).

archive $\mathcal{A}_N$ that retains all policies that have participated on the mainland (algorithm 4, line 2).

Traditional QD methods improve coverage, and thereby diversity, by uniformly sampling policies from the behavior archive and mutating them (by perturbing weights of the policy network) [30]. However, in cooperative multiagent settings, an exhaustive coverage of the behavior space is intractable and generally unnecessary: it is beneficial to focus on the regions of the behavior space that yield complementary policies capable of working in teams [11]. To facilitate this, we introduce two crucial changes. First, instead of uniform sampling, a policy θ is sampled using a biased discrete probability distribution that is a function of θ's fitness Φ on the mainland (motivated by [9]). This ensures that diversity search fluidly selects regions of the behavior space that currently produce high-fitness policies on the mainland (algorithm 4, line 3). Second, instead of using a noise-based mutation [12], NS-ES with a novelty objective ψ (equation 1) is used to systematically move N steps through the behavior space (algorithm 4, lines 4-5).

Reinforcing Class-Specific Behaviors: The experiences collected on the mainland are exploited to train a policy θ^{pg} by updating its parameters using gradient descent (similar to [19]). Over N iterations, DDPG samples random mini-batches from the buffer $\mathcal{R}_i$ and uses it to sample a policy gradient that maximizes the class-specific objective (algorithm 4, line 6). This bootstraps learning by ensuring that agents learn primitive behaviors along the physical dimension of the environment, so the mainland can attend to learning along the social (inter-agent) dimension [2].

Archiving: Following the N parallel ES and DDPG gradient updates, the trained policies θ^n and θ^{pg} are added to the elite and novelty archives (line 7).

Algorithm 4: Island

1 **Function** `island`($\mathcal{R}$: *replay buffer*, $\mathcal{A}$: *archives*):
2 $\quad \mathcal{A}^E, \mathcal{A}^N \leftarrow$ update_projection($\mathcal{R}$)
3 $\quad \theta^n \sim Categorical(\frac{\Phi(\theta_x)}{\sum_{\theta_y \in \mathcal{A}^E} \Phi(\theta_y)}) | \theta_x \in \mathcal{A}^E))$
4 $\quad$ **do in parallel for** N
5 $\quad\quad \theta^n \leftarrow$ `ES_step` (ψ, θ^n)
6 $\quad\quad \theta^{pg} \leftarrow ddpg(\mathcal{R})$
7 $\quad$ add_to_archives($[\theta^{pg}, \theta^n]$, $[\mathcal{A}^E, \mathcal{A}^N]$)

3.3 Alignment

After the mainland and the islands complete N updates, it is critical to share information between the two potentially orthogonal optimizations. Policies that were part of the e elite teams on the mainland, are added to the elite and novelty archives (algorithm 2, line 12). This addition will affect diversity search, as the biased sampling (algorithm 4, line 3) will have a higher likelihood of selecting high-fitness elites from the archive. Similarly, the aggregation of new experiences from the mainland to the buffers R_I will subsequently affect the latent space (and consequently diversity search) as it becomes part of the dataset used to train the autoencoder during the next iteration.

The weights w of the softmax function $\mu(w)_i = \frac{e^{w_i}}{\sum_{j=1}^{I} e^{w_j}}$ are updated using a gradient rule to move the distribution in the direction that maximizes the cumulative fitness f_i of each agent class $i \in I$ (algorithm 2, line 13). The entropy regularization term $\log \mu(i)$ ensures that a non-zero number of agents from each class i are part of every team, and hyperparameters α and ν represent the adaptation and regularization rates. The $(|T| - e)$ non-elite teams are replaced by new teams that are created by sampling from the elite archives $\mathcal{A}_E^I$ using updated distribution (lines 14, 15). This ensures that novel behaviors from the islands permeate the evolutionary process on the mainland.

4 EXPERIMENTAL SETUP

4.1 Cooperative Mining

We introduce cooperative mining, an asymmetric problem that builds on the exploration problems in [2, 12, 19, 23], in which agents deployed to a remote environment must prospect, extract and refine ores. The responsibilities are shared across three agent classes: 1) Rovers that can explore and mark suitable ore deposits; 2) Excavators that must extract the marked ores; and 3) Refiners that must purify the excavated ore.

Rovers can successfully mark a prospected ore deposit, if c rovers mark it simultaneously (we call c the coupling constraint). A rover is equipped with two distinct sensors: one that captures the density of ore deposits required for prospecting, and the other for capturing the density of agents around it.

$$S_{a,q} = \sum_{j \in J_q} \frac{1}{d(i,j)} \quad (2) \qquad S_{o,q} = \sum_{k \in K_q} \frac{v_k}{d(i,k)} \quad (3)$$

In equation 2, sensor S captures the density of agents of class a in quadrant q (the environment is divided into four quadrants, centered around the sensing agent; similar to [19]). J_q is the set of

agents in q within the agent's observation radius, and $d(i, j)$ is the Euclidean distance between the sensing agent i and the other agent j. Similarly, equation 3 computes the density of ore (deposits) o, in quadrant q, within the sensing agent i's observation radius. v_k the value associated with ore deposit k, will be used to compute the team fitness (equation 4).

Excavators can observe marked ore deposits and c excavators must visit them simultaneously to successfully extract them. The excavators possess the same density sensors (equations 2, 3). c **refiners** must visit extracted ores simultaneously to purify them. Refiners also use the two density sensors.

The team fitness $\phi(t)$ for a team t is given by:

$$\phi(t) = \sum_{p \in P} \prod v_p I(c, p) \tag{4}$$

In equation 4, $I(c, p)$ is an indicator function that is true if the coupling requirement c was met in refining an ore $p \in P$ and v_p is the value associated with that ore.

4.1.1 ***Agent Relationships.*** The team fitness $\phi(t)$ highlights the rich inter-class relationships required to succeed in this problem. In order to satisfy the coupling constraint c, each agent class must learn intra-class temporal and spatial dependencies. On the other hand, since the team fitness $\phi(t)$ only rewards refined ores, maximizing it requires each class to learn inter-class temporal dependencies.

4.2 Common Interest Mining

Common interest mining (CIM) builds off of cooperative mining with a drastic shift in class roles and available ores. We assume that prospecting has been completed and ores have been marked. CIM consists of three agent classes: 1) Excavators that can extract iron ore; 2) Drillers that can extract calcite ore; and 3) Refiners that must purify the extracted ore. The three classes use the density sensors (equations 2, 3) to observe ore deposits and other agents in the environment (one density sensor per ore type). Each class must also satisfy the c coupling constraint. A third kind of ore, gold, can also be mined if c excavators and c drillers visit it simultaneously. Value v_k for extracting gold is higher than the values for extracting iron and calcite.

The value differential induces a trust dilemma (commonly associated with the stag hunt [33] and Bach or Stravinsky [14]) wherein the excavators and drillers must independently decide whether to mine their respective ores or mine gold together. While the two classes can learn to individually mine their corresponding ores, this is a clearly suboptimal. However, if only one of them attempts to mine gold, the team is worse off (there are two pure-strategy Nash equilibria: one where both classes cooperate, and one where both defect [6]). This problem sets up a challenge in which the individual objectives are misaligned with the team objective.

4.3 Compared Baselines

DA-IM is primarily assessed on the quality of teams it can produce. We gauge quality based on quantitative metrics that capture both performance (team fitness in equation 4) and behavioral diversity (through expected action variance [28]). We also examine the distribution and fitness of policies in the behavior space produced by DA-IM. Finally, we highlight DA-IM's ability to systematically navigate the behavior space by inspecting a trajectory followed by the diversity search process on the islands.

The team fitness is compared against three baselines: 1) Multiagent Coevolution for Asymmetric Agents (IM), a hierarchical learning framework that separates diversity search and team optimization to produce teams with highly specialized behaviors [11]; 2) Multiagent Evolutionary Reinforcement Learning (MERL), a learning framework that leverages gradient-based and gradient-free optimization for sample efficient learning in sparse reward settings [19]; and 3) Malthusian Reinforcement Learning (MRL), which uses multiple instances of the problem simultaneously to force specialization of behaviors through adaptive populations [23].

4.4 Experimental Parameters

4.4.1 Environment. Experiments in the mining problems are conducted in a continuous 2D environment of size 100x100 units, with the episode length set to 50 steps.

State Space Agents receive a partial observation of their environment, represented by a density vector. In cooperative mining, the input state to each agent class is a vector of 16 values: 12 values that capture the density of the three agent classes (four per class, according to equation 2), and 4 density values for the ore deposits (one per quadrant, according to equation 3). Agents in common interest mining use 12 density values for the three agent classes and 12 density values (4 values for iron, calcite and gold deposits each) for the ores. The observation radius for each agent is randomly set to be between $[8, 12]$ units.

Action Space Each agent class has two continuous actions $(dx, dy) \in [-1.0, 1.0]^2$ for navigation, and an additional class-specific discrete action to mark, extract or refine an ore.

Rewards The dense reward used to reinforce class specific behaviors on the islands (section 3.2) is the inverse Euclidean distance, given by $r_d(i, t) = \frac{1}{d(i,k)}$, where $d(i, k)$ is the Euclidean distance between the sensing agent i and the closest ore k at step t. This reward allows the three agent classes to learn a "local skill" in their physical environment (navigating to an ore deposit) that can be beneficial in the broader team problem which requires coordination. In cooperative mining, the closest ore k, is the closest unmarked, marked and excavated ore for rovers, excavators and refiners respectively. In common interest mining, the closest ore k, is the closest iron and calcite ore for the excavators and drillers respectively. $r_d(i, t)$ is used by DA-IM and IM [11] on the islands and as the gradient-based reward for MERL [19].

Team fitness $\phi(t)$ (equation 4) is assigned to each team at the end of an episode. It is used by the evolutionary algorithm in DA-IM (on the mainland), IM and MERL. A linear combination of the team fitness $\phi(t)$ and the Euclidean reward $r_d(i, t)$ is used to train agents with MRL [23]. The cumulative fitness f_i used to update the weight vector w (algorithm 2, line 13) for an agent class i, is the total number of ores marked, excavated or refined by the respective classes. f_i determines the potential impact of each class in the environment and biases the team composition (via weights w) to be commensurate with this impact.

4.4.2 Learning Parameters. Trajectories of experiences generated by the evolutionary algorithm on the mainland (collected in R_I)

are used as the dataset for the autoencoder on each island. Each trajectory is a vector of state transition tuples (s_t, a_t, s_{t+1}) across an episode, where s_t and a_t are the partial observation and action at time step t. A trajectory captures an agent's preference for visiting ores and teaming with other agents. Unless stated otherwise, the parameters used by the baselines (section 4.3) are taken from their original work [11, 19, 23]. The evolutionary algorithms in DA-IM, IM and MERL use $N = 1000$ generations. Islands in DA-IM use $N = 1000$ (DDPG randomly samples N minibatches and ES uses N gradient steps: algorithm 2, lines 9-11). The adaptation and regularization rates for DA-IM and MRL are $\alpha = 1e^{-5}$ and $\nu = 0.01$ (algorithm 2, line 13). The tournament selection is done according to [29] and the parameters used for mutation are adopted from [19] (algorithm 3, lines 8-9). Mutation on the mainland is applied with a probability $m_p = 0.6$ to a fraction of the weights $m_f = 0.15$, for each policy in a team t (line 9). The actor and critic networks used by DDPG [24] on the islands are fully connected neural networks with input size 16 and 24 for cooperative and common interest mining, 3 hidden layers with ReLU activation and three output neurons.

4.4.3 Reported Metrics. For DA-IM, IM and MERL, the team with the highest fitness $\phi(t)$ at each generation is reported. Five instances of both the environments are created for MRL (islands in the MRL terminology [23]), and fitness for both the average and the highest performing team is reported. $\phi(t)$ is normalized such that a value of 1.0 indicates the successful completion of all ore deposits. We conduct 10 independent runs for each method with random seeds and report the average, with the shaded region showing 95% confidence interval. The computation requirements, stemming from the various gradient updates and evolutionary generations, vary considerably between the baselines; To make comparisons fair, the performance of teams is compared against the number of environment steps (frames).

We use the expected action variance (EAV) that captures the probability that two policies of the same class i, sampled randomly from the elite archive A_i^E, will select different actions when provided with the same state [28]. EAV is computed by calculating the total variational distance between the action distributions of policies that were part of the elite teams after training. For each agent class, the resultant EAV is a value between 0 to 1, where 0 indicates homogeneous policies (agents of a class invariably take the same action for a state), and 1 indicates that agents take different action when presented with a particular state.

5 RESULTS

5.1 Asymmetric Coordination

b We start by evaluating the performance of teams trained using DA-IM and the baselines on the cooperative mining problem. Success in this problem requires agents in a team to complete their class-specific objectives (marking, excavating and refining). The dense Euclidean reward $r_d(i, t)$ (section 4.4.1) used on the islands by the three agent classes incentivizes them to visit ores and is therefore perfectly aligned with the team objective. Figure 3 shows the fitness $\phi(t)$ on the cooperative mining problem for a coupling $c = 3$.

Teams trained with DA-IM and IM are able to refine roughly 80% of the ore deposits successfully. However, DA-IM is able to achieve higher performance with significantly less evaluation steps (over 80% in three million steps against 76% in five million steps). A collection of policies is trained on the islands in IM using an on-policy gradient method which requires the collection of experiences on both the islands and the mainland. In contrast, DA-IM exploits the experiences collected on the mainland using an off-policy optimizer (DDPG). Additionally, IM uses weight perturbation for mutation on the island in order to increase coverage in the behavior space. DA-IM on the other hand, performs informed search through the behavior space using ES. This is supported by the gradual gradient of the fitness curve for DA-IM, compared to the punctuated improvement for IM in figure 3.

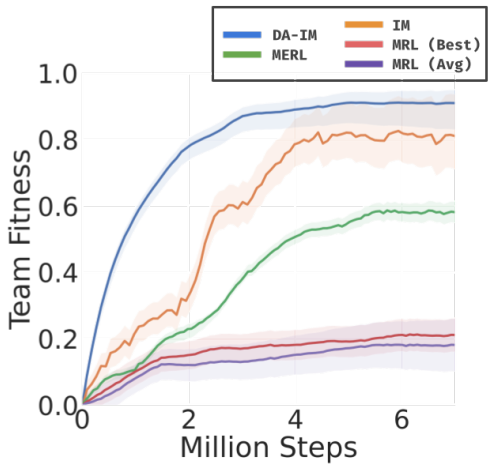

Figure 3: Team fitness on the cooperative mining problem. DA-IM outperforms the baselines while requiring significantly fewer evaluations (environment steps).

MERL uses the same rewards and the hierarchical decomposition used by DA-IM and IM. Yet it produces teams that are able to refine under 60% of the deposits. This can be attributed to the lack of diversity in teams (a hypothesis that is tested in section 5.3). Policies sampled from high-fitness teams show convergence to a homogeneous "optimal". A perturbation in the action due to the stochastisity of policies, combined with the inherent non-stationarity, can have cascading effects that break the chain of the inter-agent and intra-agent coupling required to succeed.

Teams trained with MRL are unable to consistently refine ores. Policies from teams inspected in isolation demonstrate that each agent-class is able to maximize its class-specific reward. However, MRL fails to adaptively allocate these policies across its islands (environment instances). Policies ("Species" [23]) in MRL consolidate experiences from all islands which also homogenizes them.

5.2 Asymmetric Coordination: Misaligned Objectives

Asymmetric multiagent problems are characterized by agent classes with distinct objectives or preferences that can be misaligned with the team objective [7, 25]. Hierarchical methods that independently

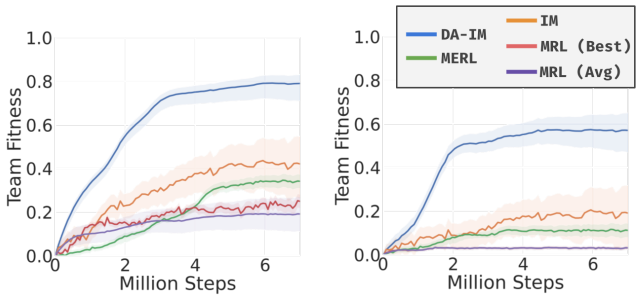

Figure 4: Team fitness on the common interest mining problem with coupling $c = 3$ (left) and $c = 6$ (right). DA-IM produces teams with diverse high-fitness policies that can effectively overcome misaligned objectives.

maximize agent-specific and team objectives must have mechanisms in place to ensure that agents are able to balance these objectives. We use the common interest mining (CIM) problem to highlight DA-IM's ability to align policies trained on individual objectives to the team objective. Crucially, the Euclidean reward $r_d(i,t)$ in CIM incentivizes excavators and drillers to visit iron and calcite ores that can contribute $v_k = 5$ per refined ore to the team fitness $\phi(t)$. However, the team objective can be maximized by excavating gold ores that have a higher value of $v_k = 20$.

Figure 4 shows the team fitness on CIM for coupling $c = 3$ (left) and $c = 6$ (right). Teams trained with DA-IM are able to refine roughly 80% of the total ores, albeit requiring more time to reach fitness comparable to cooperative mining. Initially, policies on the island that optimize the class-specific objective have a high likelihood of getting added to the elite archive successfully. During the course of learning, the ability to visit an ore, acquired from the islands, permeates the evolutionary population on the mainland (as it samples policies from the elite archive). Gradually, as policies on the mainland learn to visit the higher valued gold ores, the likelihood of a policy on the island to be added to the elite archive decreases since the behavior space is transformed to favor policies from the mainland. By exploiting the experiences generated on the mainland, the islands are able to effectively bootstrap the team fitness on the mainland further. The performance does not deteriorate substantially as the coupling is increased (figure 4, right).

The performance of teams trained with IM declines considerably on this problem since IM relies on adequate alignment between the team and class-specific objectives. Since the optimization processes on the islands and the mainland learn independently from disjoint experiences, policies transferred between the two processes are invariably rejected due to low fitness (on the mainland) or low class-specific reward (on the islands). The performance deteriorates further as coupling is increased since the likelihood of agents to discover good joint policies, merely through the evolutionary optimization on the mainland, declines.

Teams trained with MERL outperform IM on both coupling constraints. Like DA-IM, the gradient-based and evolutionary optimization processes share experiences in MERL allowing agents to bootstrap using class-specific rewards. However, the lack of diversity in MERL limits its performance. Consistent with cooperative

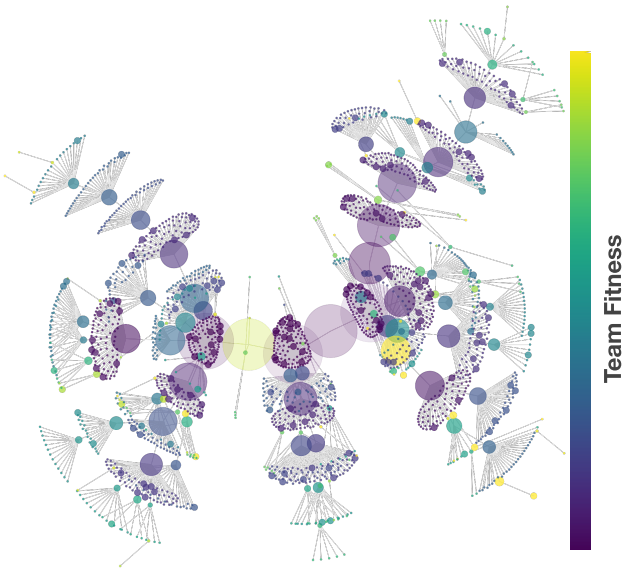

Figure 5: Phylogenetic tree demonstrating the progression of diversity search in DA-IM for rovers in the cooperative mining problem. Each node represents a policy. The node color and opacity indicate the average fitness of teams it participated in (yellow is higher). The size of a node indicates the number of descendants it produced. The tree highlights that systematic ES gradient applied to low-fitness policies can create multiple lineages of diverse (shown spatially) high-fitness descendants.

mining, MRL produces low-fitness teams with policies that usually learn to optimize the class-specific rewards, but fail to learn inter-class relationships required to succeed.

5.3 Informed Diversity

Finally, we quantitatively examine the behavioral diversity acquired between agents of the four classes in cooperative mining (CM) and common interest mining (CIM). The expected action variance (EAV) for each class is computed using the action distributions generated by evaluating the elite teams after training has been completed. Table 1 shows the EAV for DA-IM and the baselines. Due to its low fitness in the previous experiments, we disregard MRL.

Between the three compared methods, DA-IM achieves the highest EAV for all four agent classes. We also observe that the EAV is typically higher when the class-specific and team objectives are misaligned (in CIM). We hypothesize that this is likely a consequence of the drillers and excavators adaptively choosing between the two ores that are incentivized by their class-specific and team objectives. Indeed, this is supported by the distribution of policies in the inferred behavior space (consequently demonstrating the quality of the elite archive) for the driller class (figure 6, A). DA-IM produces uniform coverage in the inferred space with several high-fitness (average fitness across the elite teams) policies. Interestingly, policies from elite teams trained with IM also exhibit some diversity, albeit attaining significantly lower fitness. This can be

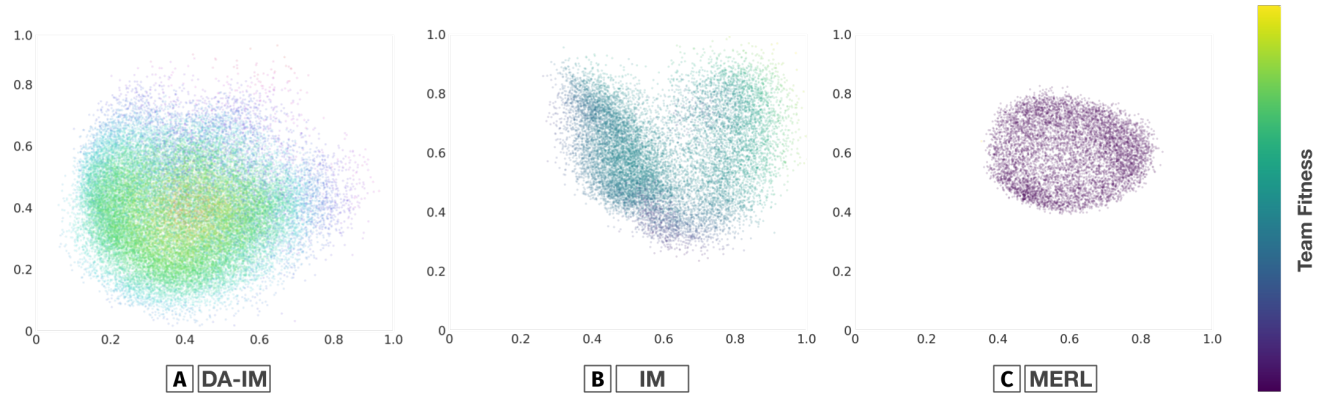

Figure 6: Inferred behavior spaces for the driller class in common interest mining. DA-IM produces diverse high-fitness policies (yellow is higher) with a uniform coverage (A). IM's behavior space is characterized by sparse disjoint regions that contain low-fitness policies (B). Policies projected in the behavior space (generated by DA-IM) show low-fitness homogeneous solutions for MERL (C).

likely attributed to IM's use of uninformed mutation on the islands, leading to policies that exhibit diverse, but low-fitness behaviors. The inferred behavior space (figure 6, B) confirms this observation: it is sparsely populated with low-fitness policies. MERL lacks an explicit diversity search mechanism and instead optimizes policies in a team to converge to a single "optimal" behavior (the parameter sharing in MERL partly contributes to this). Policies trained with MERL exhibit low EAV across the classes and environment. This is also supported by the uniform localized spread of low-fitness policies in the behavior space (figure 6, C).

DA-IM's informed diversity search can be attributed to not only the experiences from the mainland used to train the autoencoder, but also to the gradient-based ES that allows systematic exploration in the behavior space. Figure 5 highlights the progression of diversity search for rovers in the cooperative mining problem.

	CM	CIM
DA-IM		
Rovers / Drillers	0.832	0.717
Excavators	0.587	0.824
Refiners	0.270	0.311
IM		
Rovers / Drillers	0.630	0.665
Excavators	0.334	0.592
Refiners	0.283	0.298
MERL		
Rovers / Drillers	0.216	0.187
Excavators	0.239	0.296
Refiners	0.172	0.103

Table 1: Expected action variance for the four agent class on cooperative mining (CM) and common interest mining (CIM).

6 DISCUSSION

This work introduces Diversity-Aligned Island Model (DA-IM), a multiagent learning framework that enables teams of asymmetric agents to learn diverse behaviors required to balance potentially conflicting class-specific and team objectives. DA-IM leverages a combination of gradient-based and gradient-free optimizers which converge in concert by sharing policies through the behavior spaces. The gradient-free evolutionary algorithm evolves a population of teams to maximize the team objective. The experiences collected during evolution are used by each agent class to: 1) Sample policy gradients using a gradient-based off-policy algorithm to reinforce behaviors that maximize class-specific objectives; and 2) Train an autoencoder to learn low dimensional representations of the state-action trajectories drawn from the experiences. The resulting reduced spaces are utilized by a gradient-based evolution strategy to perform informed diversity.

Periodically, policies from both processes are added to the shared behavior spaces. The evolutionary algorithm samples policies from the behavior spaces thus ensuring that diverse policies from the gradient-based optimizers permeate the team population. The experiences collected during evolution are used to retrain the autoencoder which will then bias the diversity search process towards regions of the behavior space that yield high-fitness policies.

DA-IM's primary goal was to enable informed diversity search and ensure alignment between class-specific and team objectives. The separation of diversity search and team optimization opens several opportunities to parallelize and scale DA-IM to address nuanced problems that require agents to consider a spectrum of trade-offs between potentially conflicting objectives.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation grant No. NSF IIS-2112633 and Air Force Office of Scientific Research grant No. FA9550-19-1-0195.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[2] John P Agapiou, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. 2022. Melting Pot 2.0. *arXiv preprint arXiv:2211.13746* (2022).

[3] Adrian K Agogino and Kagan Tumer. 2004. Unifying temporal and structural credit assignment problems. In *Autonomous agents and multi-agent systems conference*.

[4] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.

[5] Hans-Georg Beyer and Hans-Paul Schwefel. 2002. Evolution strategies–a comprehensive introduction. *Natural computing* 1 (2002), 3–52.

[6] Hans Carlsson and Eric Van Damme. 1993. 12 Equilibrium Selection in Stag Hunt Games. *Frontiers of game theory* (1993), 237.

[7] Mai Lee Chang, Greg Trafton, J Malcolm McCurry, and Andrea Lockerd Thomaz. 2021. Unfair! perceptions of fairness in human-robot teams. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 905–912.

[8] 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

[9] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. 2018. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems* 31 (2018).

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

[11] Gaurav Dixit, Everardo Gonzalez, and Kagan Tumer. 2022. Diversifying behaviors for learning in asymmetric multiagent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 350–358.

[12] Gaurav Dixit and Kagan Tumer. 2023. Learning Inter-Agent Synergies in Asymmetric Multiagent Systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 1569–1577.

[13] 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.

[14] Gustav Feichtinger and Franz Wirl. 1993. A dynamic variant of the battle of the sexes. *International Journal of Game Theory* 22 (1993), 359–380.

[15] Manon Flageat, Bryan Lim, and Antoine Cully. 2023. Multiple Hands Make Light Work: Enhancing Quality and Diversity using MAP-Elites with Multiple Parallel Evolution Strategies. *arXiv preprint arXiv:2303.06137* (2023).

[16] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. 2020. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*. 94–102.

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

[18] Atil Iscen, Ken Caluwaerts, Jonathan Bruce, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. 2015. Learning tensegrity locomotion using open-loop control signals and coevolutionary algorithms. *Artificial life* 21, 2 (2015), 119–140.

[19] 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).

[20] 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.

[21] Guillaume J Laurent, Laëtitia Matignon, Le Fort-Piat, et al. 2011. The world of independent learners is not Markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems* 15, 1 (2011), 55–64.

[22] Joel Z Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. 2019. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742* (2019).

[23] 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.

[24] 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).

[25] Ying Liu, Brent Logan, Ning Liu, Zhiyuan Xu, Jian Tang, and Yangzhi Wang. 2017. Deep reinforcement learning for dynamic treatment regimes on medical registry data. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 380–385.

[26] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*. 6379–6390.

[27] 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

[28] Kevin R McKee, Joel Z Leibo, Charlie Beattie, and Richard Everett. 2022. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 21.

[29] Brad L Miller, David E Goldberg, et al. 1995. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems* 9, 3 (1995), 193–212.

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

[31] Mitchell A Potter and Kenneth A De Jong. 1994. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*. Springer, 249–257.

[32] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).

[33] Brian Skyrms. 2001. The stag hunt. In *Proceedings and Addresses of the American Philosophical Association*, Vol. 75. JSTOR, 31–41.

[34] 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.

[35] Kagan Tumer and Adrian Agogino. 2009. Improving air traffic management with a learning multiagent system. *IEEE Intelligent Systems* 24, 1 (2009), 18–21.

[36] Kagan Tumer, Zachary T Welch, and Adrian Agogino. 2008. Aligning social welfare and agent preferences to alleviate traffic congestion. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. Citeseer, 655–662.

[37] Karl Tuyls and Gerhard Weiss. 2012. Multiagent learning: Basics, challenges, and prospects. *Ai Magazine* 33, 3 (2012), 41–41.