Objective-Informed Diversity for Multi-Objective Multiagent Coordination

Gaurav Dixit[a,*] **and Kagan Tumer**[a]

[a]Oregon State University
ORCID (Gaurav Dixit): https://orcid.org/0000-0003-4553-8405, ORCID (Kagan Tumer): https://orcid.org/0009-0007-3809-7257

Abstract. To coordinate in multiagent settings characterized by multiple objectives, asymmetric agents (agents with distinct capabilities and preferences) must learn diverse behaviors to balance trade-offs between agent-specific and team objectives. Hierarchical methods partially address this by leveraging a combination of Quality-Diversity methods that illuminate the behavior space and evolutionary algorithms that use non-dominated sorting over the explored behaviors to improve coverage in the objective space. However, optimizing diverse behaviors and trade-offs in isolation is susceptible to producing egocentric behaviors that favor agent-specific objectives at the cost of team objectives. This work introduces the Multi-Objective Informed Island Model (MOI-IM), an asymmetric multiagent learning framework that fosters diverse behaviors and rich inter-agent relationships, necessary to balance potentially conflicting and misaligned objectives. An evolutionary algorithm improves coverage in the objective space by evolving a population of teams, while a gradient-based optimization infers and progressively explores the behavior space by fluidly adapting search to regions that produce policies with non-dominated trade-offs. The two processes are coupled via shared replay buffers to ensure alignment between coverage in the behavior and objective space. Empirical results on an asymmetric multi-objective coordination problem highlight MOI-IM's ability to produce teams that can express diverse trade-offs and robust relationships required to balance misaligned objectives.

1 Introduction

Multiagent learning is a ubiquitous and important paradigm that characterizes many real-world problems such as healthcare coordination [24], air traffic control [17, 36], and robotic automation [20, 23]. Success in such settings requires asymmetric agents (agents of distinct classes with unique capabilities and objectives) to not only learn good actions, but to learn good *joint actions* [3]. The problem is exacerbated when agents have diverse and potentially conflicting objectives that must be balanced with the team objectives [34, 21].

Quality Diversity (QD), unlike traditional optimization methods, facilitates diversity-first optimization by explicitly developing a population of diverse and high-performing policies, archived in a behavior space [30]. Exploration of behavioral diversity is maximized by genetic algorithms that repeatedly sample, mutate and catalogue policies to improve coverage in the behavior space [7]. However, exhaustive search in the behavior space is intractable and often unnecessary in multiagent settings: exploration should focus on regions of the behavior space which have the capacity to produce cooperative policies. Informed exploration is therefore pertinent for effectively learning a repertoire of diverse policies [5].

Recent developments in multiagent multi-objective QD methods have utilized hierarchical methods that make exploration tractable by transforming the behavior space into smaller class-specific subspaces [10]. However, conducting diversity search across disjoint subspaces opens the possibility for agents to learn egocentric behaviors that favor their individual class-specific objectives at the cost of team objectives. When class objectives are only aligned partially, coordination between asymmetric agent classes becomes challenging [11].

This work introduces Multi-Objective Informed Island Model (MOI-IM), a multiagent multi-objective learning framework that produces teams of asymmetric agents capable of expressing diverse trade-offs by balancing individual and team objectives. MOI-IM uses a combination of gradient-based and gradient-free optimization with shared replay buffers and behavior archives that allow them to converge simultaneously. An evolutionary algorithm evolves a population of teams (groups of policies) using non-dominated sorting to maximize team fitness and coverage of trade-offs in the objective space. The experiences collected by teams are stored in replay buffers that are utilized by a combination of gradient-based methods to: 1) train an autoencoder to infer a behavior space; 2) reinforce class-specific preferences to maximize individual objectives; and 3) perform mutation in the behavior space using an evolution strategy.

Periodically, diverse policies produced by the gradient-based optimization are added to the shared archives. The evolutionary algorithm samples policies from the shared archives to replace the low-fitness teams, thus allowing diverse policies to permeate the team population. Similarly, the experiences collected by the evolutionary algorithm will feed into the gradient-based optimizers through the shared replay buffers and directly guide the diversity search process by shaping the behavior space.

Experiments in an asymmetric multi-objective exploration problem highlight MOI-IM's ability to produce high-fitness teams that express diverse trade-offs with high coverage in the objective space and specialize when the desired trade-off is known a priori.

2 Background

Multiagent Learning A primary difficulty of learning to coordinate in a multiagent setting is the credit assignment problem: agents

* Corresponding Author. Email: dixitg@oregonstate.edu.

in a team must learn to isolate their impact on the team using a single sparse feedback [25]. Reward shaping methods partially address this challenge by decomposing the sparse feedback into distilled "stepping stone" rewards that reinforce promising individual actions which lead to sub-goals or salient events [37, 26]. However, designing shaped rewards often requires intimate knowledge of the problem, and careful consideration to avoid potential misalignment between rewards [13].

Recent advances in hierarchical learning architectures have led to the development of population-based methods that utilize a combination of gradient-based and gradient-free optimization to address the two distinct aspects, learning along the physical and social dimension, independently [18, 2]. A gradient-based method optimizes primitive agent-specific behaviors while the gradient-free method encourage cooperative planning and decision-making [19, 12]. Malthusian Reinforcement Learning (MRL), for instance, promotes behavior specialization by applying selection pressure to a population of agents that must learn primitive behaviors which are required to cooperate effectively [22]. However, in multiagent multi-objective problems, isolating the two learning dimensions can lead to egocentric behaviors that inhibit teams from expressing diverse trade-offs between potentially misaligned team objectives [10, 27].

Quality Diversity (QD) is a family of diversity-first methods that shift the traditional goal of optimizing a single behavior to maximizing the coverage of behaviors in a behavior space [30]. In its most elementary form, QD follows a two-step iterative process: 1) Sample a policy from a population of policies and mutate it; 2) Archive the mutated policy in the behavior space (using tournament if a policy already exists in that region) [5]. In spite of its success in producing a diverse repertoire of behaviors for a wide range of single-agent problems, applying it to multiagent problems remains challenging due to its reliance on the behavior space definition [31, 16]. In multiagent settings, suitable definitions such as the "inclination to coordinate" are difficult to formalize. Recent developments in QD have partially addressed this by utilizing dimensionality reduction methods to infer the behavior space from a population of policies [7]. However, effectively guiding the diversity search and behavior space inference through a sparse team objective remains challenging [10].

Evolution Strategies are a family of gradient-based methods that update a parameterised distribution over solutions in the direction that maximizes their fitness [4]. Our work uses a specific variant from this family, OpenAI-ES (henceforth abbreviated to ES), that updates the parameters of an isotropic multi-variate Gaussian distribution with mean θ and variance σ, using approximate natural gradients [35]. A population of Z solutions is evaluated on its k-nearest neighbor novelty score $\eta(\theta) = \frac{1}{K}\sum_{k=1}^{K} ||d_\theta - d_k||_2$ to get a gradient estimate that maximizes η. Algorithm 1 provides an overview of ES with the novelty-based fitness η.

Algorithm 1: ES Gradient Step (adapted from [35])

Function `ES_step(`*F: objective, θ_t: solution*`)`**:**
- ϵ_1,..., $\epsilon_Z \sim \mathcal{N}(0, I)$
- $\eta_i = \eta(\theta_t + \sigma\epsilon_i)$, for $i = 1$ to Z
- $\theta_{t+1} \leftarrow \theta_t + \lambda\frac{1}{Z\sigma}\sum_{i=1}^{Z}\eta_i\epsilon_i$
- return θ_{t+1}

3 Multi-Objective Informed Island Model

The Multi-Objective Informed Island Model (MOI-IM) is a multi-agent learning framework that produces teams of cooperative asymmetric agents (agents with different capabilities and preferences) that exhibit diverse trade-offs to balance individual and team objectives.

Figure 1 presents a high-level outline of MOI-IM. An island i is initialized for each agent class with a unique utility function u_i which describes a class's preferences: a scalarization that indicates how the class values each team objective. An island is also assigned a randomly initialized population of policies pop_i, a replay buffer $\mathcal{R}_i$ and two empty archives $\mathcal{A}_i^E$ and $\mathcal{A}_i^N$ (algorithm 2, lines 1-4).

At a high-level, MOI-IM progresses as follows: M teams of S agents each, are created by sampling from populations $pop_{i\in I}$ using a categorical distribution with probabilities given by a softmax μ over weights w_I (algorithm 2, line 6). The I weights over the islands dictate the team composition and will be adapted as learning progresses. The T teams are evolved on the mainland using an evolutionary algorithm and experiences collected during their evaluation are stored in replay buffers $\mathcal{R}_I$ (line 7). The islands will use these experiences to: 1) train an autoencoder to infer a behavior space; 2) reinforce class-specific preferences; and 3) conduct informed diversity search (line 11). The weights w are updated using a gradient rule to shift team composition in the direction that maximizes the cumulative fitness f_i for each agent class (line 12). Finally, e elite teams are retained and the remainder are replaced by sampling teams from the elite archives $\mathcal{A}_{i\in I}^E$ (lines 13-14). This process is repeated until a convergence criterion is met (adequate coverage in the objective space or sufficiently high team fitness for a given trade-off — section 3.2.2).

Algorithm 2: Multi-Objective Informed Island Model

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

3.1 Mainland: Team Optimization

The goal of the mainland process is to evolve teams of agents (groups of policies) to maximize coverage of trade-offs in the objective space. This is achieved by adapting key components from NSGA-II [9]. In each generation, teams $t \in T$ are evaluated on team objectives and their experiences are collected in replay buffers $\mathcal{R}_{i\in I}$ (algorithm 3, lines 2-3). The fitness vector Φ_t for each team $t \in T$ represents a trade-off in the team objectives. Teams are sorted into Pareto fronts using non-dominated sorting and crowding distance using Φ_t (lines 4-5). e highest-fitness teams from the top fronts are retained for the

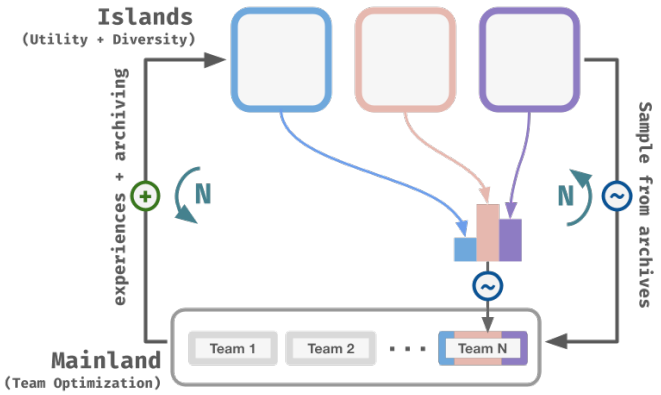

Figure 1. MOI-IM Overview: Each island is a combination of a gradient-based optimization, behavior space inference and diversity search process for an asymmetric agent class (figure 2). A mainland represents an evolutionary optimization over a population of teams (groups of policies), that are evolved using non-dominated sorting and crowding distance to maximize coverage of trade-offs in the objective space. After N learning updates, the experiences collected on the mainland are used on the islands to progress diversity search, infer an updated behavior space, and reinforce class-specific objectives. Diverse policies from the islands are then sampled on the mainland using a softmax distribution μ to inject diversity in the mainland evolutionary optimization.

next generation as elites (line 6). If a Pareto front has more than e teams, the crowding distance is used to break ties.

Algorithm 3: Mainland (NSGA-II, adapted from [9, 10])

1 **Function** `mainland` (T: *teams*, $\mathcal{R}_I$: *Replay Buffers*) **:**
2 **for** *generation* $\leftarrow 0$ **to** N **do**
3 $\Phi_t, \mathcal{R}_I = evaluate(t) \mid \forall t \in T$
4 $T \leftarrow sort_\Phi(T)$ `// Non-dominated sort`
5 $C \leftarrow crowding\_distance(T)$
6 $E = T[0:e]$ `// top e teams using` C
7 Create set $S = \emptyset$, using binary tournament:
8 **while** $|S| < (|T| - e)$ **do**
9 $t \leftarrow crossover(t_x \sim U(E), t_y \sim U(T - E))$
10 $t \leftarrow mutate(t)$ `// perturb weights`
11 $S \leftarrow S \cup t$
12 $T \leftarrow S \cup E$
13 $add\_to\_archives(T[0:e]_i, [\mathcal{A}_i^E, \mathcal{A}_i^N]) \quad \forall i \in I$

Binary tournament is used to apply mutation and crossover to the policies from the remainder of the teams (algorithm 3, lines 7-11). A single-point crossover is performed by uniformly sampling policies from low-fitness and elite teams to create new policies that have a higher likelihood of succeeding [32]. Weights of the new policies are perturbed by applying Gaussian noise (equivalent to a polynomial mutation [8]). The mainland, over N generations, will gradually improve the fitness of teams and coverage in the objective space.

3.2 *Islands: Informed Diversity*

An island is a set of optimization processes for each agent class that reinforces class-specific preferences, infers and performs search through the behavior space. Figure 2 presents an outline of an island.

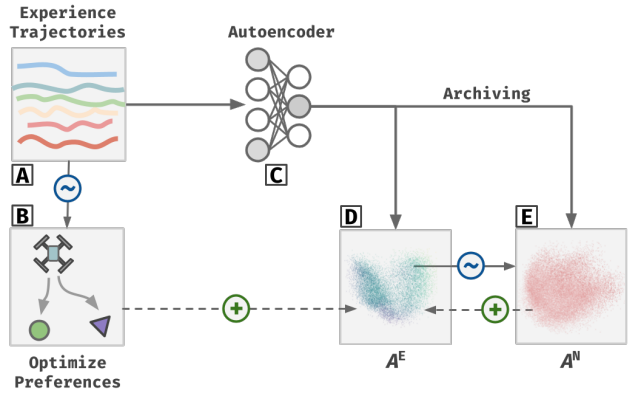

Figure 2. Island Overview: Trajectories of experiences collected on the mainland are used as a dataset to train an autoencoder (A-C). The resulting latent space is used as the behavior space for archiving elite ($\mathcal{A}^E$) and novel ($\mathcal{A}^N$) behaviors. Experiences are sampled by a policy gradient to reinforce class-specific preferences (A-B). An evolution strategy samples a policy from the elite archive ($\mathcal{A}^E$) and takes N gradient steps in the novelty archive ($\mathcal{A}^N$) to maximize its k-neighbor novelty score η (D-E). Finally, policies from (B) and (E) are added to the elite archive which will be used to sample policies into teams on the mainland.

3.2.1 *Behavior Space Inference*

The trajectories of experiences collected during the evolutionary process on the mainland in replay buffers $\mathcal{R}_{i \in I}$, are used as a dataset on each island to train an autoencoder [7]. The dimensionality reduction results in a latent space that captures the variance in policies of each class [1]. The latent space is used as a behavior space to conduct diversity search [10, 7]. An island i maintains two behavior space instances: an elite archive $\mathcal{A}_i^E$ that catalogues high-fitness policies that are currently participating in the evolutionary process on the mainland, and a novelty archive $\mathcal{A}_i^N$ that retains all policies that have historically participated on the mainland (algorithm 4, line 2).

Algorithm 4: Island

1 **Function** `island` ($\mathcal{R}_i$: *replay buffer*, $\mathcal{A}_i$: *archives*) **:**
2 $\mathcal{A}_i^E, \mathcal{A}_i^N \leftarrow update\_projection(\mathcal{R}_i)$
3 $\theta^n \sim \chi$ `// strategy` χ `from section 3.2.2`
4 **do in parallel for** N
5 $\theta^n \leftarrow$ `ES_step` (η, θ^n)
6 $\theta^{pg} \leftarrow ddpg(\mathcal{R}_i)$
7 add_to_archives($[\theta^{pg}, \theta^n]$, $[\mathcal{A}_i^E, \mathcal{A}_i^N]$)

3.2.2 *Informed Diversity Search*

Traditional QD methods improve coverage in the behavior space by uniformly sampling and mutating policies [33]. However, exhaustive coverage of the behavior space is intractable in multiagent settings and often unnecessary: it is pertinent to focus on specific regions of the behavior space that yield policies capable of working cooperatively in teams [10]. We introduce two crucial changes that facilitate an informed search. First, instead of uniform sampling, a policy θ is sampled from a biased probability distribution that underscores θ's contribution to the teams on the mainland (algorithm 4, line 3). We consider several sampling strategies, χ, to assess θ's contribution:

1. Uniform Sampling: On an island i, a policy θ is uniformly sampled from the elite archive $\mathcal{A}_i^E$. This strategy encourages uniform coverage in the behavior space and is traditionally used by QD methods [33].

$$\chi \sim Uniform(\mathcal{A}) \quad : \mathcal{A} = \mathcal{A}_i^E \tag{1}$$

2. Uniform Sampling from the Pareto front: policy θ is uniformly sampled from the subset $\mathcal{A}_{PF}$ of the elite archive $\mathcal{A}_i^E$, which only contains policies from teams on the Pareto front. This ensures that policies in high-fitness teams are the focal point of diversity search (motivated by [39], which suggests that search should be focused on the elite hypervolume).

$$\chi \sim Uniform(\mathcal{A}_{PF}) \quad : \mathcal{A}_{PF} \subseteq \mathcal{A}_i^E \tag{2}$$

3. Biased Sampling: θ is sampled from a categorical distribution that favors policies in the elite archive $\mathcal{A}_i^E$ that maximize the preferences (utility κ_i) of agents on island i. This ensures that diversity search fluidly considers policies that are currently part of high-fitness teams and maximize the preferences of the agent class.

$$\chi \sim Categorical(\frac{\kappa_i(\theta_x)}{\sum_{\theta_y} \kappa_i(\theta_y)}) \quad | \, \theta_x, \theta_y \in \mathcal{A}_i^E \tag{3}$$

$$\kappa_i(\theta) = \sum_{j \in J} \lambda_j \frac{1}{|T|} \sum_{t \in T} \Phi_j(t) \tag{4}$$

$\kappa_i(\theta)$ is the weighted scalarization of the J objectives Φ, averaged over the T teams in which the policy θ participated. The preferences of agents on island i are given by weights λ.

4. Chebyshev Sampling: Finally, we consider the case when the goal is to evolve teams of agents to maximize fitness for a desired trade-off Z that is known a priori. The weighted scalarization κ_i in equation 4 is replaced by the Chebyshev method scalarization $\varkappa_i$:

$$\varkappa_i(\theta) = max_{j \in J} \{\lambda_j \frac{1}{|T|} \sum_{t \in T} | \, \Phi_j(t) - Z_j \, |\} \tag{5}$$

The Chebyshev scalarization $\varkappa_i$ computes the maximum weighted deviation from the desired trade-off Z (where Z_j is the desired value of the j-th objective) [38]. This encourages diversity search to actively explore diverse behaviors that can minimize the deviation from the desired trade-off.

Second, instead of applying Gaussian or polynomial mutation (contrast to [10]), NS-ES with a novelty objective is used to take N gradient steps towards a new policy θ in the behavior space (algorithm 4, lines 4-5). The two changes ensure a systematic search through the behavior space on each island, which adapts to the progress made on the mainland.

3.2.3 Reinforcing Class-Specific Preferences

Experiences collected in the replay buffers R_I on the mainland are exploited on the islands to train a policy θ^{pg} using Deep Deterministic Policy Gradient (DDPG) [23]. On each island i, DDPG samples mini-batches from the corresponding replay buffer R_i and uses them to sample a policy gradient that maximizes a scalarized dense reward function (algorithm 4: line 4, 6). This has the benefit of training agents with primitive class-specific behaviors which maximize their preferences (given by weights λ), that can then be utilized for learning higher-level cooperative behaviors on the mainland [2].

3.3 Alignment and Archiving

As the mainland and the islands serve unique and potentially orthogonal optimization roles, it is pertinent that information be shared between them to ensure aligned optimization. This is achieved by sharing replay buffers and behavior archives.

Following N generations of evolution on the mainland, policies from the elites teams on the mainland are added to the corresponding elite and novelty archives on each island (algorithm 3, line 13). These high-fitness policies will be sampled on the islands subsequently (with a higher likelihood for biased sampling strategy χ). Moreover, the experiences collected on the mainland will shape the behavior space (algorithm 4, line 2) and thus directly influence NS-ES. After N gradient updates on the island, policies θ^n and θ^{pg} are added to the elite and novelty archive (algorithm 4, line 7).

Weights w of the softmax function $\mu(w)_i = \frac{e^{w_i}}{\sum_{k=1}^{I} e^{w_j}}$ are updated using gradient ascent to update the team sampling distribution in the direction that maximizes the total fitness $f_{i \in I}$ of each agent class (algorithm 2, line 12). The entropy regularization $\log \mu(i)$, controlled with the regularization rate ν, ensures that a non-zero number of agents from each class (island) participate in teams. To introduce the progress made on the islands to the mainland, $(|T| - e)$ lowest fitness teams are replaced by sampling policies from the elite archives $\mathcal{A}_{i \in I}^E$ into new teams using μ (lines 13-14).

The use of a distinct elite and novelty archive on each island is crucial: policies on an island i are sampled from the elite archive $\mathcal{A}_i^E$, whereas the NS-ES steps are performed in the novelty archive $\mathcal{A}_i^N$ to ensure that novel policies are indeed novel from the policies that have already participated on the mainland previously [14].

4 Experimental Setup

Multi-Objective Cooperative Mining: We introduce cooperative mining, an asymmetric bi-objective problem that builds on motifs from several exploration challenges [2, 10, 19, 22]. Agents in the cooperative mining problem are deployed to a remote environment where they must cooperatively prospect, extract and refine ores. Mining responsibilities are shared between three agent classes: 1) Rovers that must first prospect and mark ores suitable for extraction; 2) Excavators that can then extract marked ores; and 3) Refiners that must then purify the mined ores. Deposits of two distinct ores, iron and calcite, are distributed uniformly throughout the environment.

Rovers can successfully mark an ore deposit, if c rovers visit it simultaneously (we call c the coupling constraint). Similarly, c excavators are required to mine an ore and c refiners to purify it. Agents are equipped with two distinct sensors: one that captures the density of ore deposits, and the other to capture the density of agents in their observation radius.

$$S_{o,q} = \sum_{k \in K_q} \frac{v_k}{d(i,k)} \tag{6} \qquad S_{a,q} = \sum_{j \in J_q} \frac{1}{d(i,j)} \tag{7}$$

In equation 6, sensor S captures the density of an ore o (iron or calcite) in quadrant q (the observation is dividend into four quadrants, centered around the sensing agent i; motivated by [19]), within the sensing agent i's observation radius. Each ore deposit k has a value v_k associated with it for successfully mining and purifying it. Similarly, equation 7 computes the the density of agents of class a in quadrant q. J_q is the set of agents in quadrant q within a's observation radius, and $d(i,j)$ is the Euclidean distance between the sensing agent i and the other agent j.

Team Fitness: Cooperative mining has two objectives: purifying iron and calcite ores. The fitness of a team Φ_t is a vector, formally defined as:

$$\begin{cases} \phi_0 = \sum_{k \in K_i} \prod v_k I(c,k) & \\ \phi_1 = |K_c| \cdot e^{\frac{-|K_c|}{\psi}} & K_c \ \forall k \in K_c : I(c,k) == 1 \end{cases} \tag{8}$$

In equation 8, the reward ϕ_0 for purifying an iron ore $k \in K_i$ increases linearly while the reward ϕ_1 for calcite ore $k \in K_c$ is modeled as a curve that plateaus after ψ calcite ores have been purified. $I(c,k)$ is an indicator function that is set to true if c excavators or drillers visited the ore k simultaneously, followed by c refiners.

Class-Specific Utilities: A utility function u_i for each agent class i defines their preference for visiting iron and calcite ores.

$$u_i = \lambda_0 \cdot \sum_{k \in K_i} v_k + \lambda_1 \cdot \sum_{k \in K_c} v_k \tag{9}$$

The utility function u_i is used on the islands as a dense reward to reinforce class-specific primitive behaviors (such as independently visiting an ore since u_i is not dependent on the coupling constraint c) using DDPG (section 3.2.3). The weights λ for each class specify preferences that are also used for sampling from the elite archive on each island (equations 3, 5)

Agent Relationships: The team fitness Φ_t (equation 8) underscores the rich inter and intra-class dependencies required to succeed in the cooperative mining problem. Agents must learn to maximize both the team objectives, ϕ_0 and ϕ_1, and their class-specific utilities. The coupling constraint c enforces intra-class dependencies, while the ordered marking, extraction and refining imposes strong inter-class dependencies.

4.1 Compared Baselines

MOI-IM's primary goal is to produce cooperative teams that can express diverse trade-offs. To that end, we gauge the quality of teams based on their fitness (equation 8), coverage of trade-offs in the objective space, adaptation in team composition, and the behavioral diversity measured using expected action variance [28]. We also empirically evaluate the effect of various sampling strategies (section 3.2.2) by examining the distribution of policies in the inferred behavior space. Finally, we investigate if informed diversity search can lead to high-fitness regions in the behavior space by inspecting the gradient steps taken by diversity search using a phylogenetic tree.

The team fitness, Pareto fronts and action variance is compared with four baselines that address different dimensions of the multi-agent multi-objecitve problem: 1) NSGA-II, a widely used multi-objective evolutionary algorithm that leverages non-dominated sorting and crowding distance to produce Pareto fronts with high fitness and coverage [9]; 2) SPEA2, a traditional multi-objective optimization method that explicitly archives and ranks solutions using density estimates in the objective space [40]; 3) Multi-objective Asymmetric Island Model, an island model-based multi-objective learning framework that improves coverage of trade-offs in the objective space by training teams on a wide variety of tasks simultaneously [10]; and 4) Malthusian Reinforcement Learning (MRL), a hierarchical learning framework that encourages agents to specialize by applying selection pressure over various tasks (called islands) [22].

4.2 Experimental Parameters

4.2.1 Environment

The cooperative mining problem is instantiated as a continuous 2D environment of size 100x100 units, with uniformly distributed ore deposits and an episode length of 50 time-steps.

State Space: The state for each agent is a partial observation within their observation radius. The observation is a vector of 20 density values: 12 values that capture the density of the three agent classes (densities in the four quadrants for each class; equation 7) and eight density values for the ore deposits (four for iron and calcite each; equation 6). The observation radius for each agent is randomly sampled to be between [10, 14] units. Each ore deposit also has an observation radius between [4, 12] units: c agents have to be within a deposit's radius to perform their class-specific task (equation 8).

Action Space: Agents have two continuous navigational actions $(dx, dy) \in [-1.0, 1.0]^2$, and an additional discrete class-specific action (available in an ore's radius) to mark, extract and refine an ore.

Rewards: On the islands, the class-specific utility, equation 9, is used as the dense reward by the policy gradient in MOI-IM and on the islands by MO-AIM [10]. The utility function (parameterized by the weights λ and ore values v_k) reinforces each class' preferences by rewarding an agent for simply visiting an ore independently. The weights λ_0 and λ_1 for rovers, excavators and refiners are set to $[0.7, 0.3]$, $[0.2, 0.8]$ and $[0.5, 0.5]$ respectively. These weights can be interpreted as follows: Iron deposits have a larger observation radius and are therefore easier to find and mark. Therefore, rovers have a higher utility for marking iron deposits. Excavators prefer extracting the calcite ores, while refiners are indifferent to the ore type. The values v_k for deposits are sampled from $[2, 8]$ and the maximum number of calcite ores to refine ψ is set to 8.

The team fitness vector Φ_t (equation 8) is assigned to the teams at the end of each episode. It is used by MOI-IM and MO-AIM on the mainland, and by NSGA-II and SPEA2 as the fitness of candidate solutions. MRL uses a linear combination of the team fitness and class-specific utility to train policies across its islands [22]. The cumulative class fitness f_i used to update the parameters of the distribution μ (algorithm 2, line 12) is the summed value, across the teams on the mainland, of the ores that were marked, excavated and refined by the three classes respectively. f_i isolates the impact each class had in the teams on the mainland.

4.2.2 Learning Parameters

Replay buffers R_I store trajectories of the experiences which is used as the dataset for the autoencoder (parameters from [7]) on each island. A trajectory is a vector of state transitions $(s_t, a_t, s_{t+1}, u_{i,t})$ encountered by a policy as it participates in a team on the mainland. Note that the utility $u_{i,t}$ is computed at each time step t but is only used on the islands when DDPG samples from the replay buffer. The actor and critic networks on each island [23] are fully connected neural networks with input size 20, 3 hidden layers with ReLU activation, and three output neurons.

Unless stated otherwise, baselines use parameters from their original work [10, 22, 7, 9]. The evolutionary algorithms in MOI-IM and MO-AIM use $N = 1000$ (number of generations and gradient steps; algorithm 2, lines 9-11). The adaptation and regularization rates used for updating μ, are $\alpha = 1e^{-5}$ and $\nu = 0.01$ (algorithm 2, line 12). Binary tournament on the mainland is done according to [29] and the mutation parameters are adopted from [19] (algorithm 3, line 10).

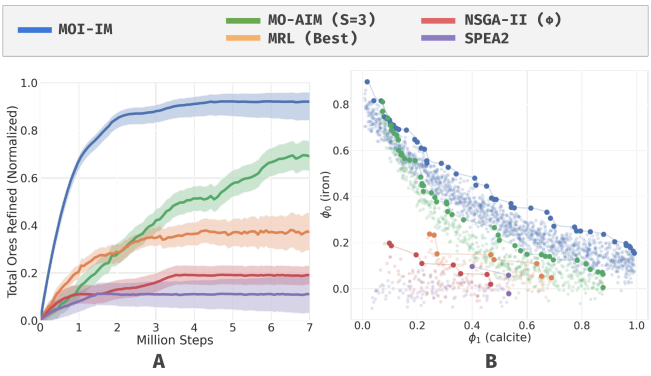

Figure 3. Number of Ores refined (A) and the Pareto fronts produced (B) by teams of 16 agents in the cooperative mining problem with coupling $c = 3$. MOI-IM outperforms all baselines significantly while requiring fewer evaluations.

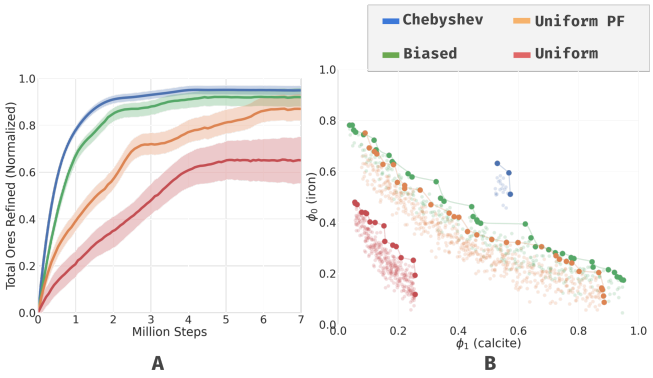

Figure 4. Number of Ores refined (A) and the Pareto fronts produced (B) by teams of 18 agents in the cooperative mining problem with coupling $c = 4$. When the desired trade-off is known a priori, Chebyshev sampling allows MOI-IM to concentrate diversity search on a small subset of the behavior space, leading to high-fitness teams in under a million evaluations. When the desired trade-off is not known, biased sampling produces the highest coverage in the objective space.

Mutation on the mainland is applied to each policy in a team by perturbing a fraction of the weights $m_f = 0.15$ with Gaussian noise with a probability $m_p = 0.3$ (congruent to the mutation probability and index of a polynomial mutation [8]).

4.2.3 *Reported Metrics*

For all baselines, the highest number of successfully refined ores (normalized) at each generation is reported. Three instances of the cooperative mining environment are created for the islands in MRL and the mainlands in MO-AIM [11]. 10 independent runs are conducted for each baseline with random seeds. The average and 95% confidence interval is shown for both the fitness (shaded region) and the Pareto front (dominated trade-offs shown with smaller alpha). The computation requirements across the methods differ significantly: MRL requires training across multiple instances of the problem, while MO-AIM requires several optimization processes to run in concert. To make comparisons fair, all metrics are reported against the total number of environment steps (frames).

5 Results

5.1 *Multi-Objective Coordination*

We begin by examining the performance of teams, measured using the number of ores refined successfully, and the coverage of trade-offs in the objective space. Figure 3 compares MOI-IM's performance with the baselines in the cooperative mining problem with a coupling $c = 3$.

Teams trained with MOI-IM refine over 85% of the ore deposits successfully and in significantly fewer learning steps (over 70% in under 1.5 million against 7 million steps for MO-AIM). MO-AIM's higher variance and slower learning can be attributed to two reasons. First, MO-AIM's on-policy gradient optimization requires independent evaluations on both the islands and the mainlands. In contrast, the islands in MOI-IM exploit experiences collected on the mainland. Second, a crucial aspect of diversity search is the mutation strategy. In MO-AIM, this is achieved by perturbing weights with Gaussian noise. MOI-IM on the other hand, performs systematic search through the behavior space which is informed by the performance of policies of the mainland (supported in section 5.2, figure 6).

Examining the Pareto fronts suggests a potential correlation between the behavior space exploration strategy and the resultant diversity in trade-offs (figure 3.B; partially supported in [15]). While the best teams trained with MO-AIM succeed to refine over 75% of the deposits, the performance is limited to a subset of the Pareto front. In contrast, MOI-IM yields high-fitness teams that exhibit significant coverage in the objective space.

Teams trained with MRL are unable to consistently refine ores. The selection pressure applied on MRL's islands encourages specialization of roles which discourages the rovers and the excavators to cooperate against their utility functions. Indeed, on an island on which the rovers learn to specialize, we see them consistently satisfy c in order to mark deposits. However, the misaligned preferences with the excavators restricts inter-class cooperation that is required to succeed on islands on which both classes participate.

Traditional multi-objective optimization methods NSGA-II and SPEA2 do not inherently account for asymmetric agents in a team. Therefore, we use the distribution μ learnt by MOI-IM to sample teams for them. Both methods fail to produce high-fitness teams, which can be attributed to the misaligned preferences, lack of an explicit (behavioral) diversity mechanism, and fitness sparsity due to the coupling requirement.

5.2 *Informed Diversity*

In multiagent multi-objective settings, agent must learn diverse behaviors that allow them to balance their individual preferences with the team objectives [34]. As exhaustive search through the behavior space is intractable and unnecessary [10], it is pertinent to consider strategies that aid in effective exploration of the behavior space. We consider several different sampling strategies χ (section 3.2.2) that dictate the trajectory ES will take to navigate in the behavior space. Figure 4 highlights the effect of χ on the average team performance in the cooperative mining problem with coupling $c = 4$.

5.2.1 *Decision Support Setting*

In these settings, the desired trade-off is available only after learning is complete and is subject to change. Therefore the goal is to

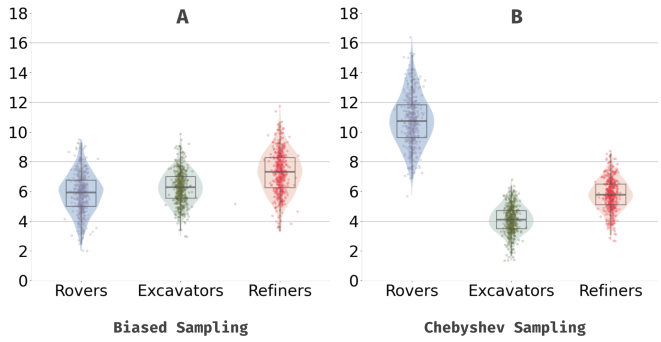

Figure 5. Biased sampling produces teams which contain roughly equal number of rovers, excavators and refiners. Chebyshev sampling allows μ to adapt for the desired trade-off Z, with a higher number of rovers.

maximize coverage in the objective space. We start with the traditional QD strategy of sampling from the behavior space (elite archive $\mathcal{A}^E$) uniformly [30]. Uniform sampling does not specifically attend to regions that produce promising policies in order to favor exhaustive search. Finding suitable behavioral diversity thus takes longer (figure 4.A) and results in a sparsely explored behavior space which produces low-fitness teams.

Next, instead of sampling uniformly from the elite archive $\mathcal{A}^E$, we sample from the policies in the archive that are on the Pareto front (Uniform PF in figure 4). This substantially improve both learning speed (sub-figure A) and the coverage in the objective space (sub-figure B). Sampling from the Pareto front allows diversity search to concentrate on regions of the behavior space that yield diverse trade-offs between the two objectives. Our hypothesis is that selective sampling from the Pareto front is closely related to searching in the elite hypervolume [39].

We build on this further by incorporating a categorical distribution that favors policies in the elite archive which maximize the individual utility function (motivated by [6]). This allows diversity search to consider diverse policies which have a high utility but are not on the Pareto front yet. We notice marginal improvements in objective space coverage and overall team performance. However, the learning speed improves substantially (sub-figure A) as islands are able to fluidly adapt search to promising regions in response to the evolutionary optimization on the mainland.

5.2.2 *Known Trade-offs*

This is a special case in which the desired trade-off Z is known a priori. The goal shifts from maximizing coverage in the objective space, to finding high-fitness teams that can express Z. We replace the linear scalarization κ in biased sampling with the Chebyshev scalarization $\varkappa$ (section 3.2.2). Intuitively, Chebyshev sampling favors policies from teams that are the furthest away from Z, and encourages ES to explore regions of the behavior space that can minimize the deviation from Z. In the cooperative mining problem with coupling $c = 4$, 18 agents, and 8 ore deposits of each type uniformly distributed, we set Z to perfectly balance the fitness from both objectives ϕ_0 and ϕ_1 (note that the precise value of Z changes for each run because it is a function of v_k: equation 8).

Figure 4 shows the average performance of teams and the coverage using Chebyshev sampling. Of the four strategies evaluated, learning with Chebyshev sampling is the fastest across the 15 independent seeds. By focusing utility optimization to regions of the

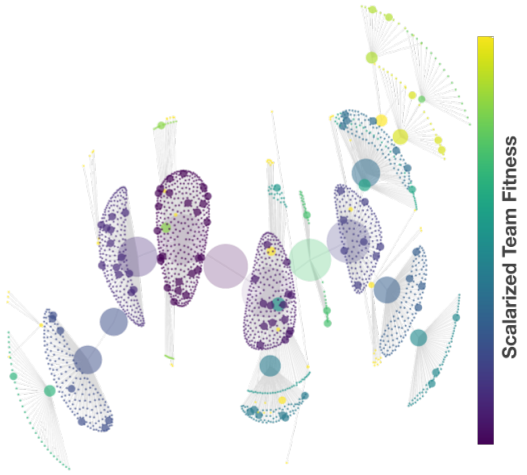

Figure 6. A phylogenetic tree that highlights the trajectory taken by diversity search for rovers in the cooperative mining problem. Each node represents a policy produced by ES that was added to the elite archive. The node size represents the number of descendants the policy produced (through mutation on the mainland and sampling on the island). The node color indicates the average scalarized fitness of the teams in which it participated (yellow is higher). The phylogenetic tree shows that systematic application of ES to low-fitness policies can create distinct lineages of high-fitness and diverse (shown spatially) descendants.

behavior space that deviate from the trade-off, the islands are able to drive search towards regions that can achieve the desired trade-off Z, at the cost of minimal coverage in the objective space. This is also evident in the drastic difference between team composition (across 15 seeds), as shown in figure 5. The diversity produced using Chebyshev sampling is also reflected by the team sampling distribution μ, which adapts to sampling a higher number of rovers in teams (sub-figure B). This is beneficial since the desired trade-off Z favours rovers that can work against their individual preference and mark both iron and calcite ore deposits equally.

6 Discussion

This work introduces Multi-Objective Informed Island Model (MOI-IM), a multiagent multi-objective learning framework that produces teams of asymmetric agents capable of expressing diverse trade-offs by balancing their class-specific and team objectives. MOI-IM leverages an evolutionary algorithm that uses non-dominated sorting with crowding distance to evolve high-fitness teams that improve coverage in the objective space. The experiences collected by the evolutionary algorithm are used to reinforce class-specific objectives, shape the behavior space, and guide an evolution strategy through the behavior space. Periodically, diversity is injected into the evolutionary population by replacing teams with policies sampled from the behavior space. The shared replay buffers and behavior spaces ensure that diversity search is guided by the team objective optimization.

MOI-IM's diversity search considers several sampling strategies that can fluidly adapt search to regions of the behavior space where progress is currently being made. However, these strategies assume that class-specific scalarization is fixed and known a priori (equation 3). In future work, we will consider sampling strategies that are agnostic to class-specific preferences and allow further alignment between diversity search and the team objectives.

Acknowledgements

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

[2] J. P. Agapiou, A. S. Vezhnevets, E. A. Duéñez-Guzmán, J. Matyas, Y. Mao, P. Sunehag, R. Köster, U. Madhushani, K. Kopparapu, R. Comanescu, et al. Melting pot 2.0. *arXiv preprint arXiv:2211.13746*, 2022.

[3] S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[4] H.-G. Beyer and H.-P. Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1:3–52, 2002.

[5] C. Colas, V. Madhavan, J. Huizinga, and J. Clune. Scaling map-elites to deep neuroevolution. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, Jun 2020. doi: 10.1145/3377930.3390217. URL http://dx.doi.org/10.1145/3377930.3390217.

[6] E. Conti, V. Madhavan, F. Petroski Such, J. Lehman, K. Stanley, and J. Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31, 2018.

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

[8] K. Deb and D. Deb. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, 4(1):1–28, 2014.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[10] G. Dixit and K. Tumer. Learning synergies for multi-objective optimization in asymmetric multiagent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 447–455, 2023.

[11] G. Dixit and K. Tumer. Learning inter-agent synergies in asymmetric multiagent systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1577, 2023.

[12] G. Dixit, E. Gonzalez, and K. Tumer. Diversifying behaviors for learning in asymmetric multiagent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 350–358, 2022.

[13] H. C. L. Duc Thien Nguyen, Akshat Kumar. Credit assignment for collective multiagent rl with global rewards. In *Advances in Neural Informaiton Processing Systems (NIPS 2018): Montreal, Canada, December 2-8*, pages 8102–8113, 2018.

[14] M. Flageat, B. Lim, and A. Cully. Enhancing map-elites with multiple parallel evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1082–1090, 2024.

[15] M. C. Fontaine, J. Togelius, S. Nikolaidis, and A. K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 94–102, 2020.

[16] A. Gaier, A. Asteroth, and J.-B. Mouret. Automating representation discovery with map-elites. *arXiv preprint arXiv:2003.04389*, 2020.

[17] J. Hill, J. Archibald, W. Stirling, and R. Frost. A multi-agent system architecture for distributed air traffic control. In *AIAA guidance, navigation, and control conference and exhibit*, page 6049, 2005.

[18] A. Iscen, K. Caluwaerts, J. Bruce, A. Agogino, V. SunSpiral, and K. Tumer. Learning tensegrity locomotion using open-loop control signals and coevolutionary algorithms. *Artificial life*, 21(2):119–140, 2015.

[19] S. Khadka, S. Majumdar, and K. Tumer. Evolutionary reinforcement learning for sample-efficient multiagent coordination. *CoRR*, abs/1906.07315, 2019. URL http://arxiv.org/abs/1906.07315.

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

[21] J. Z. Leibo, E. Hughes, M. Lanctot, and T. Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*, 2019.

[22] J. Z. Leibo, J. Perolat, E. Hughes, S. Wheelwright, A. H. Marblestone, E. Duéñez Guzmán, P. Sunehag, I. Dunning, and T. Graepel. Malthusian reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 1099–1107, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

[23] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[24] Y. Liu, B. Logan, N. Liu, Z. Xu, J. Tang, and Y. Wang. Deep reinforcement learning for dynamic treatment regimes on medical registry data. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 380–385. IEEE, 2017.

[25] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390, 2017.

[26] P. Mannion, S. Devlin, J. Duggan, and E. Howley. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review*, 33:e23, 2018. doi: 10.1017/S0269888918000292.

[27] M. Märtens and D. Izzo. The asynchronous island model and nsga-ii: study of a new migration operator and its performance. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1173–1180, 2013.

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

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

[30] J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[31] J. Nordmoen, K. O. Ellefsen, and K. Glette. *Combining MAP-Elites and Incremental Evolution to Generate Gaits for a Mammalian Quadruped Robot*, pages 719–733. 03 2018. ISBN 978-3-319-77537-1. doi: 10.1007/978-3-319-77538-8_48.

[32] W. Paul, C. William, and K. De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. *Journal Spector*, page 15, 2002.

[33] J. K. Pugh, L. B. Soros, and K. O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3: 40, 2016.

[34] R. Rădulescu, P. Mannion, D. M. Roijers, and A. Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10, 2020.

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

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

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

[38] K. Van Moffaert, M. M. Drugan, and A. Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pages 191–199. IEEE, 2013.

[39] V. Vassiliades and J.-B. Mouret. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 149–156, 2018.

[40] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.