# Learning Synergies for Multi-Objective Optimization in Asymmetric Multiagent Systems

Anonymous Author(s)
Submission Id: pap734s2

## ABSTRACT

Agents in a multiagent system must learn diverse policies that allow them to express complex inter-agent synergies required for teamwork. Multiagent Quality-Diversity methods partially address this by transforming the agents' large joint policy space to a tractable sub-space that can produce synergistic agent policies. However, in multi-objective problems with asymmetric agents (agents with different capabilities and objectives), the search for diversity is fundamentally guided by the need to learn a Pareto front of policies that represents diverse trade-offs between agent-specific and team objectives. This work introduces Multi-objective Asymmetric Island Model (MAIM), a multi-objective multiagent learning framework for the discovery of generalizable agent synergies and trade-offs via adaptation of population dynamics over a spectrum of tasks. The key insight is that the competitive pressure arising from the changing populations on the team tasks forces agents to acquire robust synergies required to balance their individual and team objectives in response to the nature of their teams and task dynamics. Results on several variations of a multi-objective habitat problem highlight the potential of MAIM in producing teams with diverse specializations and trade-offs that readily adapt to unseen tasks.

## CCS CONCEPTS

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

## KEYWORDS

Team Composition, Quality Diversity, Multiagent learning

**ACM Reference Format:**

Anonymous Author(s). 2018. Learning Synergies for Multi-Objective Optimization in Asymmetric Multiagent Systems. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Multiagent learning is a promising paradigm that has shown success in a wide variety of real-world problems such as air traffic control [8], robotic automation [10, 13] and healthcare coordination [14]. Interestingly, a majority of these applications rely on asymmetric agents (agents with different capabilities and objectives) to not only learn good joint actions, but also learn robust *inter-agent synergies* which are crucial for adaptation to changes in the team and task dynamics. The problem is further aggravated when multiple objectives must be optimized simultaneously.

Recent work in Quality Diversity (QD) approaches has paved the way towards learning a repertoire of diverse policies instead of optimizing one optimal policy [18, 20]. The shift towards diversity-first methods can potentially allow asymmetric agents to discover several complementary policies which are conducive to cooperative team behaviors. Traditional QD methods operate by explicitly populating the space of policies so as to maximize coverage. However, exhaustively working through the policy space is largely intractable for multiagent systems since the joint policy space grows exponentially with the number of agents.

Multiagent Coevolution for Asymmetric Agents (MCAA), a recently proposed multiagent framework, partially addresses the difficulty of exploring a joint policy space by filtering it into smaller sub-spaces that yield good teaming policies [6]. However, the filtering is driven by the performance of agents on a single team objective which often leads to over-specialization of agent synergies and behaviors. In multi-objective settings, it is crucial to learn generalizable inter-agent synergies that can foster diverse trade-offs between team and agent-specific objectives in order to learn a Pareto front with sufficient coverage in the objective space [21].

This work introduces Multi-objective Asymmetric Island Model (MAIM), a multiagent learning framework that produces teams of asymmetric agents capable of learning diverse inter-agent synergies and team trade-offs that can be generalized across a variety of tasks. MAIM combines Quality Diversity optimization and multi-objective coevolutionary optimization with a migration strategy that allows both processes to converge to a diverse set of policies that can balance agent-specific and team objectives in order to work together as robust teams.

The Quality Diversity process enables a population of agents to learn diverse primitive behaviors that maximize their agent-specific utility (preferences towards objectives). The coevolutionary optimization on the other hand, concurrently evolves a population of teams (groups of agents) to find Pareto fronts that balance the team objectives in the objective space. Periodically, policies from the QD process are migrated to replace the policies from the lowest fitness teams thereby injecting diversity in the team population, whereas policies from teams on the Pareto front are migrated to the QD process to bias its search towards regions of the policy space conducive to good team behaviors.

A softmax distribution guides the allocation of policies from the QD to the coevolutionary process and is updated after each migration so as to maximize the cumulative agent utility across the team tasks. *The competitive pressure arising from the changing distribution of asymmetric agents across the team tasks forces agents*


*Woodstock '18, June 03–05, 2018, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

*to acquire generalizable inter-agent synergies that allow agents to exercise diverse trade-offs between agent-specific and team objectives in response to the dynamics of the task, team and agent behaviors.*

Experiments in a multi-objective asymmetric habitat problem highlight the potential of MAIM in producing teams that learn diverse strategies and specializations to balance the agent-specific and team objectives across a spectrum of seen and unseen tasks.

## 2 RELATED WORK

*2.0.1* ***Multiagent Learning.*** A key challenge of learning in multiagent settings is credit assignment: agents must learn to access their own contribution in a team using a sparse team feedback [11, 24]. Reward shaping techniques can address this by transforming a team reward to "stepping stone" rewards [16]. However, this requires intimate knowledge of the problem and is susceptible to creating misaligned rewards [7]. Techniques like Multiagent DDPG use a centralized-learning-decentralized-execution paradigm in which a single learner optimizes the joint policy [15]. Although this can make learning tractable, it is often difficult to scale to asymmetric multiagent settings since agents have distinct action spaces and individual objectives.

Evolutionary and population based gradient-free methods offer a promising solution since they are particularly suitable for learning with sparse feedback. Multiagent Evolutionary Reinforcement Learning (MERL) in particular, combines the strengths of gradient based policy and gradient free evolutionary optimization to learn in cooperative settings [9]. By evolving policies trained on agent-specific behaviors to maximize team fitness, MERL implicitly selects for alignment. However, the shared policy and population architecture limits MERL to symmetric multiagent settings.

*2.0.2* ***Quality Diversity.*** Quality Diversity (QD) are a family of methods that shift the focus from optimizing one policy to discovering a diverse repertoire of policies [17]. In its most basic form a QD process can be described as a two step iterative process: 1) Mutate a policy from a population of policies; and 2) Catalogue the mutated policy in the population policy space, selecting for the higher fitness policy in case of replacement [3]. A major challenge in applying QD to complex problem with unknown variables, which is typical of multiagent problems, is its reliance on a definition of the policy space. Recent methods have addressed this by using a dimensionality reduction method to infer the policy space [4].

However, scaling this effectively to multiagent systems remains challenging due to the large policy space that is a resultant of agents' policies being largely dependent on each other. Recent advances to alleviate this include Multiagent Coevolution for Asymmetric Agents (MCAA), Malthusian Reinforcement Learning (MRL) and Minimum Criteria Coevolution (MCC), which effectively transform the policy space into smaller tractable sub-spaces by means of population dynamics (MCAA and MRL) or resource limitation (MCC) [2, 6, 12]. While these methods are able to discover diverse agent policies, they are difficult to scale to multi-objective settings which require agents to learn trade-offs between individual and team objectives. Our method bridges this gap by transforming the policy space into a tractable space that can produce policies with diverse trade-offs.

## 3 MULTI-OBJECTIVE ASYMMETRIC ISLAND MODEL

Multi-Objective Asymmetric Island Model (MAIM) is a multiagent framework that trains teams of asymmetric agents (agents with different objectives and capabilities) to balance their individual (potentially conflicting) objectives with the team objectives on a wide set of tasks. MAIM produces a set of "islands" that allow agents to maximize their individual objectives and a set of "mainlands" on which teams of agents are evolved to balance team objectives. Figure 1 presents an overview of MAIM.

Each island in MAIM hosts a population of agents of a unique class that share a utility function (conspecific utility), which is a scalarization function that maps the team reward to a scalar that describes the class's preferences: how the class values each team objective. A Quality-Diversity process with an unstructured archive [4, 6] is carried out on each island to enable the local population to learn diverse policies that maximize the island's conspecific utility. Figure 2 provides an overview of the island process.

A mainland in MAIM represents a unique team task that demands a specific Pareto set of team behaviors that can balance the objectives associated with it. Each mainland hosts a population of teams (group of policies) that is trained to learn a Pareto front of team behaviors (joint policies of the agents in the team) using a coevolutionary algorithm.

Policies from the islands are periodically sampled, using a softmax distribution $\mu_i$ associated with each island $i$, to replace the policies in the worst performing teams on the mainlands. $\mu_{i \in I}$ on each island governs the allocation of its population to the mainlands. The policy migration allows teams on mainlands to incorporate

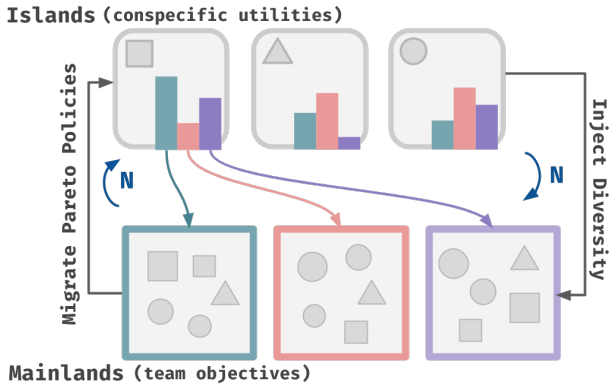


**Figure 1: Each agent class is assigned to an island. Agents learn diverse primitive behaviors that maximize their agent-specific utilities using Quality Diversity (figure 2). Each mainland (represents a unique team task) evolves a population of teams to learn a Pareto front that balances the objectives of that mainland. Every $N$ iterations, policies from the Pareto front teams are migrated from the mainlands to the islands to bias the population on the islands towards diversity that is conducive to good team behaviors. Similarly, policies from the islands replace the worst performing teams on the mainlands, which results in increased diversity on the mainlands.**

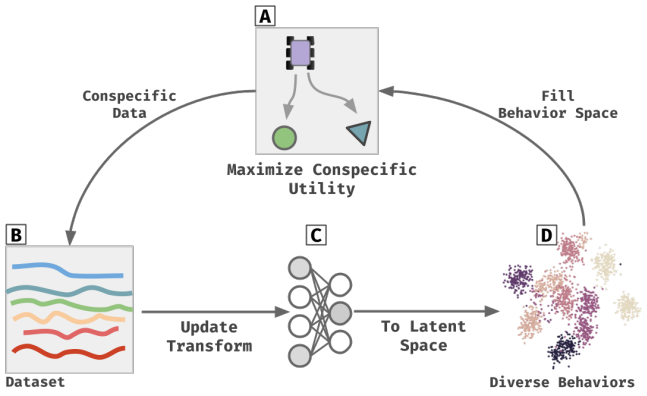


**Figure 2: Quality-Diversity optimization on an Island. Agents are trained to optimize a conspecific utility (A). Data collected from the training (B) is used to train a dimensionality reduction method (C). The resultant latent space acts as a behavior space that is filled by mutating policies on the island (D).**

diverse policies discovered on the islands. Likewise, policies on the mainlands from the best performing teams (on the Pareto front) are migrated to the islands to bias their diversity search towards regions of the policy space that produce good team behaviors.

The softmax distribution on each island is updated after every migration, via a gradient rule, to maximize the expected conspecific utility of agents on that island across all mainlands (team tasks). This update ensures that the allocation of island policies to the mainlands is commensurate to their relative performance on the mainlands. Each mainland has a fixed carrying capacity (maximum number of policies that can reside on it) which forces the island populations to learn complementary policies that can work together with other agents classes across the spectrum of tasks (mainlands).

*3.0.1 Conspecific Utility Optimization on Islands.* The islands follow a QD approach (adapted from [4]): Each island hosts a population of policies (neural networks initialized with random weights) for a specific agent class. A softmax distribution $\mu_i$ is initialized with a weight vector $\omega_i$ for each island. A QD process (described next) for each island runs in parallel for $N$ iterations (algorithm 1, lines 2-3). A random policy $\pi$ is sampled from the population $pop_i$ (line 4) and is mutated (Gaussian perturbation applied to the weights of the neural network; line 5) to create a new policy $\pi\prime$ on island $i$. A rollout is conducted for $\pi\prime$ in the environment which generates conspecific data $\tau$ and a reward vector (with a reward value for each objective) $r$ (line 6). The reward vector is then scalarized using the conspecific utility $u_i()$, which is used to update the weights of $\pi\prime$ using PPO [23] (lines 7-8). The updated policy is added to the island population and its corresponding conspecific data $\tau$ is added to the island dataset (lines 9-10). Any suitable data that summarizes an agents behavior in the environment can be used as the conspecific data. Prior works have successfully used a variety of metrics including agents' trajectory, end effector positions and agent speeds to infer a latent policy space [4, 6]. Conspecific data used in our experiments is specified in section 4.3.2.

After the $N$ QD iterations, the dataset on each island is used to re-train a dimensionality reduction method that produces a reduced latent space which is used as the policy space for the subsequent island iterations (algorithm 1, line 12). The policies are then projected in the updated latent space (using $\tau$), and policies with high conspecific utility are retained in case of overlaps [4].

**Algorithm 1:** Islands (optimize conspecific utilities)

```
1  Function run_islands(pop_I:|I| Populations):
2      for iteration ← 0 to N do
3          Do in parallel for island i ∈ I
4              sample policy π ∈ pop_i
5              π′ = mutate(π)              // perturb policy weights
6              τ, r = rollout (π′)         // local rollout
7              r_u ← u_i(r)                // compute utility from reward vector
8              update π′ (τ, r_u) using PPO // update policy π′
9              pop_i ← pop_i ∪ π′          // add to island population
10             dataset_i ← dataset_i ∪ (τ, π′)
11     foreach island i ∈ I do
12         train_DR(dataset_i)             // update latent space
13         foreach policy (τ, π) ∈ pop_i do
14             project(τ)                  // project to latent space
```

*3.0.2 Pareto Fronts on the Mainlands.* Each mainland hosts a unique team task. Initially, policies from the islands are sampled and assigned to the mainlands (using $\mu_{i...I}$) to create a population of teams on each mainland. A team $t$ on mainland $m$ is a set of $t_n$ policies created by randomly grouping policies assigned to $m$. Random grouping ensures that the team composition on a mainland is reflective of the relative proportion of island policies on it.

Teams on the mainlands are evolved for $N$ generations using a standard coevolutionary algorithm [19] with the selection criterion adapted from Non-dominated Sorting Genetic Algorithm (NSGA-II) [5]. The goal on each of the mainlands is to learn a Pareto Front of teams (groups of policies) that maximize the objectives on that mainland. Teams are evaluated in the environment and assigned a team reward vector $\Phi$ (a vector of scalar team rewards for each objective) (algorithm 2, lines 4-6). The teams are sorted using Non-dominated sorting [5] by projecting their corresponding reward vectors $\Phi_{t...T}$ in the objective space and the pair-wise crowding distances are computed (lines 7-8). The top $e$ teams from the dominating fronts are stored as elites $E$ (by prioritizing crowding distance in cases where an entire front cannot be included in $E$; line 9). This ensures that at the end of a generation, the teams with the highest objectives and coverage in the objective space are retained for the next generation. The policies from the non-elite ($|T| - |E|$) teams are then subjected to crossover with policies from the elites using binary tournament and mutation (Gaussian perturbation to weights) to create new teams (lines 10-14).

*3.0.3 Migrations.* The conspecific utility maximization on the islands and the Pareto front optimization on the mainlands have distinct functions and largely run in parallel. To leverage the benefits of both, it is pivotal to exchange information between the

**Algorithm 2:** Mainlands (learn Pareto fronts)

1 **Function** run_mainlands($T_M$:$|M|$ *populations of* $|T|$ *teams*):
2  **Do in parallel for mainland** $m \in M$
3  $T \leftarrow T_m$ // Teams for mainland $m$
4  **for** *generation* $\leftarrow 0$ **to** $N$ **do**
5  **foreach** *team* $t \in T$ **do**
6  $\Phi_t$ = evaluate $(t)$
7  $T \leftarrow$ rank $(T)$ // Non-dominated sort using $\Phi_{0:T}$
8  $C \leftarrow$ crowding_distance $(T)$
9  $E = T[0:e]$ // select top e elite teams using $C$
10  Select the remaining $(|T| - e)$ teams from $T$, to form set $S$ using binary tournament selection
11  **while** $|S| < (|T| - e)$ **do**
12  /* apply crossover and mutation operators */
$\pi_{off} \leftarrow$ crossover($\{(\pi_x, \pi_y) \mid \pi_x \in E, \pi_y \in S, (\pi_x, \pi_y) \in I\}$)
13  $\pi_{off} \leftarrow$ mutate $(\pi_{off})$
14  $S \leftarrow S \cup \pi_{off}$
15  $T \leftarrow S \cup E$

**Algorithm 3:** Multi-Objective Asymmetric Island Model

1 Initialize $I$ islands, one island per agent class
2 Initialize a population $pop_i$ of policies $\pi$ for each $i \in I$
3 Initialize $M$ Mainlands, one per team task
4 **Function** MAIM(*I:Islands, M:Mainlands*):
5  **for** $k \leftarrow 0$ **to** $\infty$ **do**
6  **do in parallel**
7  $Pop_I$ = run_islands $(Pop_I)$ // conspecific $u_i$...$I$
8  $T_M$ = run_mainlands $(T_M)$ // Pareto fronts
9  **foreach** *island* $i \in I$ **do**
10  $Pop_i \leftarrow Pop_i \cup T_{m,i}[0:e])\ \forall m \in M$
11  $w_{k+1,i} \leftarrow$ update$(w_{k,i})$ // according to eqn (1)
12  **foreach** *mainland* $m \in M$ **do**
13  /* Replace $(|T| - e)$ teams by sampling islands */
$T_m \leftarrow T_m[0:e] \cup (|T| - e) \sim w_{k+1,i}, \forall i \in I$

two processes. Migration is done asynchronously after every $N$ iterations on the islands and the mainlands. Policies from the $E$ elite teams from each mainland are added to the populations on islands (algorithm 3, lines 9-10). The migrated policies will participate in the following island rollouts (algorithm 1, lines 4-6), provide conspecific data (line 10) and influence the latent space inference (line 12) to bias QD to search in regions of the policy space that yield successful teaming policies.

The softmax distribution for each island $\boldsymbol{\mu(m,i)} = \frac{e^{w_{m,i}}}{\sum_{j-m} e^{w_{j,i}}}$ is then updated using a gradient rule (equation 1) to move in the direction that maximizes the expected conspecific utility $(u_{m,i})$ across the mainlands (algorithm 3, line 11).

$$\omega_{k+1,i} = \omega_{k,i} + \alpha \left[ \sum_{m=1}^{|M|} \nabla_w \mu(m,i)(u_{m,i} - v log\mu(m,i)) \right] \quad (1)$$

In equation 1, $\omega_{k,i}$ is the weight vector for the softmax distribution $\mu(m,i)$ on island $i$, for iteration $k$ (algorithm 3, line 5). $\alpha$ and $v$ are the adaptation and regularization rates. $u_{m,i}$ is the cumulative expected conspecific utility of $pop_i$ on mainland $m$. To ensure that at least a small non-zero number of policies from each island participate on the mainlands, we introduce entropy regularization $log\mu(m,i)$. This prevents early over-specialization [6] and also ensures that agents learn generalizable policies that can work across several seen and potentially held-out tasks.

Finally, the updated distribution is used to allocate policies from the islands to replace the policies in the worst performing $(|T|-|E|)$ teams on the mainlands (algorithm 3, line 13). This replacement allows the teams on the mainlands to incorporate diversity from the islands. *The combination of diversity search with conspecific utility maximization and team Pareto front optimization, allows MAIM to yield teams of asymmetric agents that learn to balance their (potentially conflicting) conspecific utilities with the team objectives.*

## 4 EXPERIMENTAL SETUP

We conduct three experiments to inspect the team performance, discovered Pareto fronts, and diverse agent synergies acquired with MAIM: 1) **Asymmetric Coordination** to evaluate team performance across five unique bi-objective scenarios that call for diverse agent synergies; 2) **Adaptation to Held-out Tasks** for accessing MAIM's ability to learn agent synergies and trade-offs that can be generalized to unseen tasks; and 3) **Agent and Team Objective Trade-offs** to examine changing relationships and trade-offs between agents in response to a changing team task.

### 4.1 Multi-Objective Habitat Problem

We introduce the multi-objective asymmetric habitat problem that builds off of design motifs from several cooperative multiagent problems (rover exploration [6], Allelopathy and Clamity [12]). Agents of three unique classes are deployed in a remote environment to conduct pre-mission activities for setting up a habitat. The agent classes are: 1) Rovers with vision sensors; 2) Excavators with digging equipment; and 3) Aerial drones with communication capabilities. The agents must work together in teams to find different grades of dig sites, excavate them and communicate the number and grade of excavated sites back to a ground station.

**Dig sites** are graded as either coarse-grained $K_c$ or fine-grained $K_f$. Gradation is an important indicator of properties like compressibility which dictate the value of a dig site in the habitat mission.

**Rovers** are equipped with a sensor that captures the presence of other agents within their observation radius (eqn 2) and a sensor that captures the presence of marked and unmarked dig sites around them (eqn 3). To successfully mark a site, $c$ rovers (referred to as the coupling requirement) must visit the site simultaneously [6].

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

In equation 2, $S_{a,q,i}$ provides the density of agents of class $a$ in quadrant $q$ of the environment (we use four quadrants, centered around the agent) for agent $i$; $J_q$ is the set of agents of class $a$ in $q$, within the agent's observation radius, and $d(i, j)$ is the Euclidean distance between agent $i$ and another agent $j \in J_q$.

In equation 3, $S_{a,q,i}$ gives the density of dig sites of class $a$ (coarse or fine) in quadrant $q$, within agent $i$'s observation radius. $v_k$ represents a scalar value associated with dig site $k$. Finally, $d(i, j)$ is the Euclidean distance between agent $i$ and a dig site $k \in K_q$.

The **Excavators** are equipped with two density sensors: one for capturing the density of agents around them (eqn 2) and the other for capturing the density of marked dig sites in their observation radius (eqn 3). Like the rovers, $c$ excavators must visit a marked site simultaneously in order to successfully excavate it.

The **Drones** are responsible for communicating any excavated sites back to the ground station: the team fitness (eqn 4) will only take into consideration excavated sites that are within a drone's observation radius. Like the excavators, drones have a sensor for computing agent densities and one for measuring dig site densities.

*4.1.1 Team Fitness.* The habitat problem has two objectives: excavating coarse $K_c$ and fine $K_f$ dig sites. The fitness of a team $\Phi_t$ is a vector of size 2 with values corresponding to scalar rewards for excavating $K_c$ and $k_f$. Formally:

$$\begin{cases} \phi_0 = \sum_{k \in K} v_k & K = \{k \in K_c | C_{(c,k)} \cdot O_k\} \\ \phi_1 = |K| \cdot e^{\frac{-|K|}{\psi}} & K = \{k \in K_f | C_{(c,k)} \cdot O_k\} \end{cases} \tag{4}$$

In equation 4, the reward $\phi_0$ for excavating coarse sites $K_c$ increases linearly while the reward $\phi_1$ for fine sites $k_f$ is modeled as a smooth curve that plateaus after excavating a maximum number of sites $\psi$. $C_{(c,k)}$ is an indicator function set to true if $c$ excavators visited the site $k$ simultaneously, and $O_k$ is an indicator that is true if site $k$ was within the observation radius of a drone.

*4.1.2 Conspecific Utilities.* The conspecific utility for each agent class is a linear weighted sum that dictates the class's preferences for visiting coarse and fine dig sites.

$$u_i = w_0 \cdot \sum_{k \in K_c} v_k + w_1 \cdot \sum_{k \in K_f} v_k \tag{5}$$

Equation 5 give utility $u_i$ for an agent of class $i$. The class specific weights $w_0$ and $w_1$ used in our experiments are specified in 4.3.1. The conspecific utility acts as a dense reward for each agent class that allows it to learn diverse primitive behaviors, such as navigating to a dig site, on the islands (algorithm 1, line 7).

*4.1.3* ***Agent Relationships and Trade-offs.*** The multi-objective fitness $\Phi_t$ captures a rich constellation of dependencies and trade-offs: On the highest level, teams of agents must be able to optimize both the team objectives ($\phi_0, \phi_1$), while simultaneously maximizing their preferences for those objectives (equation 5). On a secondary level, this balancing act is further influenced by the agent synergies required to coordinate: $c$ rovers must mark a dig site simultaneously (spatial intra-agent coupling) to make it available for $c$ excavators (temporal inter-agent coupling) which can then dig it simultaneously (spatial intra-agent coupling). This is followed by the drones (temporal inter-agent coupling) that need to team with other agents to ensure that excavated sites are considered by the team fitness (indicator $O_k$ in equation 4). *To learn and maintain these synergies, agents must learn to balance their utilities with the team objectives, and more subtly with the utilities of other agent classes.*

## 4.2 Compared Baselines

The primary metric to gauge the benefits of MAIM is the Pareto front of team policies. Pareto dominating policies are conducive to high team fitness and they implicitly require the agents to learn diverse synergies in order to successfully cooperate on a wide set of scenarios in the habitat problem. We also empirically evaluate the coverage of the Pareto front in the team objective space by testing the generalizability of the acquired agent synergies to an unseen task that requires them to adapt their conspecific utilities and team fitness trade-offs (section 4.1.3).

We compare the Parento front discovered by MAIM against several variants of traditional multi-objective and multiagent learning methods. Each baseline addresses a particular aspect of the problem that MAIM intends to solve: 1) Multiagent Coevolution for Asymmetric Agents (MCAA), an island model based framework for discovering specialized inter-agent synergies on cooperative tasks [6]; 2) NSGA-II, a standard evolutionary algorithm for optimizing multiple objectives by explicitly selecting for Pareto dominating solutions with wide coverage in the objective space [5]; 3) Malthusian Reinforcement Learning (MRL), a reinforcement learning framework that promotes agent specialization via shifting population dynamics; and 4) SPEA2, a multi-objective optimization method that archives and ranks candidate solutions by using a density estimate in the objective space [26].

MAIM combines the seemingly orthogonal strengths of these baselines: It employs the combination of diversity search and optimization via an island model (like MCAA) to concurrently optimize conspecific and team utilities, with a migration policy that allows team fitness to guide diversity search. It allows agents to discover inter-agent synergies by applying selection pressure via changing population dynamics (like MRL), and adopts non-dominated sort-based selection (like NSGA-II) within a coevolutionary algorithm to allow teams to discover Pareto fronts with wide coverage. The resultant composition of these design choices produces MAIM: a multi-objective multiagent learning framework for the discovery of generalizable agent synergies and trade-offs that foster teamwork.

## 4.3 Experimental Parameters

*4.3.1 Habitat Environment.* The habitat environment is a continuous 2D space of size 60x60 units, unless specified otherwise.

**Inputs** The input to the drones and excavators is a vector of 20 values: four density values (one per quadrant) for each agent class (eqn 2) and four values each for marked coarse and fine dig sites (eqn 3). The rovers have four additional values for unmarked coarse and fine dig sites, making their input a vector of 28 values (eqn 3). The observation radius for each agent is sampled from $\sim [5, 8]$ during initialization.

**Action Space** Agents have two navigational actions $(dx, dy) \in [-2.0, 2.0]^2$. A dig site is marked when $c = 3$ rovers visit it and is excavated when $c = 3$ excavators visit simultaneously.

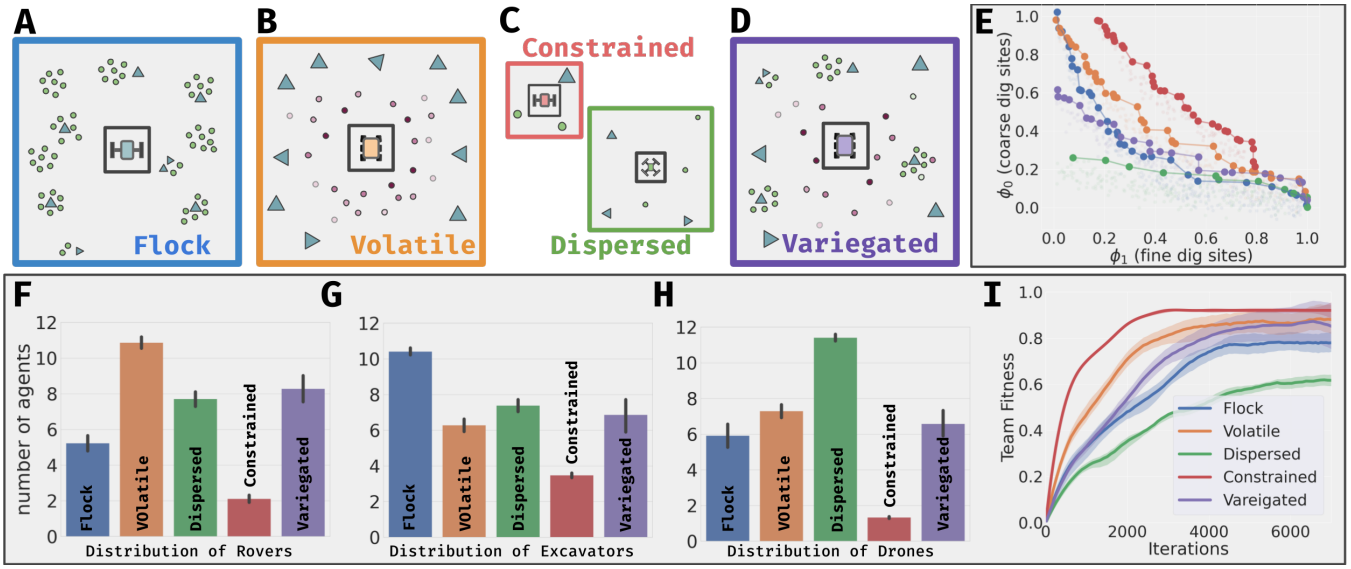


**Figure 3: MAIM is trained in concert across five scenarios (Top row (A-D)). (E) shows the Pareto fronts discovered by MAIM on each of the scenarios. The bottom row shows the softmax distributions $\mu$ that allocate agents to the scenarios (F-H) and the average team fitness across them (I). The experiment highlights MAIM's ability to compose teams with diverse agent synergies and trade-offs that generalize across a wide set of tasks. Section 5.1 gives further insights into the performance and discovered inter-agent relationships.**

**Rewards** The conspecific utility is the dense agent-specific reward that allows agents to learn primitive navigational behaviors and is used by the QD process on all islands (5. An agent gets this reward for simply visiting a dig site independently. The weights $w_0$ and $w_1$ for rovers, excavators and drones are set to $[0.3, 0.7]$, $[0.8, 0.2]$ and $[0.5, 0.5]$ respectively. For agents of class $i$ on a mainland $m$, the sum of their conspecific utilities is used as the cumulative utility $u_{m,i}$ to update the softmax distribution that dictates their allocation to mainlands (eqn 1). The value of a dig site $v_k$ is sampled from $\sim [1, 5]$ and the maximum number of fine sites to excavate is set to $\psi = 10$. Team fitness $\Phi$ (eqn 4) is used by NSGA-II, SPEA2, MRL and the mainlands on MCAA and MAIM.

*4.3.2 Learning Parameters.* The **conspecific data** $\tau$collected on the islands is a dataset of $(d_{a,t}, d_{k,t})$ vectors for episodes of length 60. $d_{a,t}$ is the distance to the closest agent $a$ and $d_{k,t}$ is the distance to the closest dig site at time $t$. For the habitat problem, this vector captures an agent's inclination to team and visit dig sites. The dataset on each island is used to train PCA ([1]; parameters from [4]) to produce a latent space, used as the behavior space, for QD.

MAIM and MCAA use $N = 1000$ policy updates on the islands and mainlands between migrations. For fair comparison, MRL, NSGA-II and SPEA2 use $2N$ updates for every $N$ updates in MAIM. An iteration $i$ in section 5 therefore corresponds to $N$ updates for MAIM, MCAA and $2N$ updates for the other baselines. MAIM, MCAA and MRL use $\alpha = 1e-05$ and $\nu = 0.01$ for updating their softmax distributions (equation 1).

# 5 RESULTS

## 5.1 Asymmetric Coordination

We start by evaluating MAIM on five distinct scenarios in the habitat problem, each of which requires learning a unique team composition, a Pareto front that can produce teams that maximize both objectives (coarse and fine dig sites) and diverse agent synergies that are generalizable across the five scenarios. Figure 3 presents the 5 scenarios (top row A-D), the learned Pareto fronts for the scenarios (E), the average team fitness (I) of the $e = 5$ best performing elite teams (algorithm 2, line 9) and the distribution $\mu$ of agents on the five scenarios (F-H).

In the **first scenario, "Flock"** (figure 3.A), the dig sites are concentrated in clusters, distributed uniformly across the environment. Furthermore, this scenario is biased towards fine dig sites $K_f$ as they are twice as likely to be generated as coarse sites $K_c$. This is a relatively easy task for rovers since their conspecific utility prefers $K_f$ (section 4.3.1) and it is widely accessible in clusters. A small number of rovers are able to optimally mark sites as is evident from their low distribution on this scenario (F). Excavators on the other hand have a higher preference towards coarse sites $K_c$ and they seem to be the most important class in this scenario (G), allowing the teams to balance both objectives fairly well (E) and excavate over 80% of the dig sites (I).

The **second scenario, "Volatile"** (figure 3.B), presents two challenges: 1) marked sites stay marked for a limited duration of 8 steps), after which they must be marked again by the rovers; 2) coarse sites $K_c$ surround fine sites $K_f$ which are concentrated in the center. To balance both objectives and perform optimally as teams in this setting, we see a significant increase in the number of

rovers in teams (average 11 rovers per team compared to 6 drones and excavators; F-H). Empirically inspecting the behaviors shows that rovers spread out in the environment with some specializing in re-marking inner $K_f$ dig sites while some venturing outwards to mark $K_c$. On an average, teams are able to excavate over 80% of the sites (I) and the spread-out strategy seem to improve the overall balance of objectives as evident in the Pareto front (E).

In the **third scenario, "Constrained"** (figure 3.C, top-left), we shrink the environment to $25x25$ units. As expected, the team size on this scenario reduces with a uniform team composition containing all three agent classes (average 2 agents of each class; F-H). An easier environment also allows MAIM to find a dominating Pareto Front that allows teams to maximize both objectives (E).

The **fourth scenario, "Dispersed"** (figure 3.C, bottom-right) doubles the environment size without changing the number of dig sites. This necessitates agents form strong synergies in order to complete the mark-excavate-communicate inter-agent coupling. Drones become the premier class in this scenario (H) and learn to team up with rovers and drones, often keeping them in their observation radius in order to capture any successfully excavated sites. The overall team performance in this setting is the lowest (I) with the Pareto front showing a higher bias towards fine dig sites (E). We hypothesize that both excavators and drones heavily rely on rovers in this scenario to lead them to marked sites, which shifts the overall objective maximization slightly towards the preference (conspecific utility) of the rovers (E).

The **fifth scenario, "Variegated"** (figure 3.D), combines motifs from Volatile (limited duration dig sites with the higher valued coarse sites $K_c$ spread away from the center) and Flock (site clusters). The distribution $\mu$ for each class indicates a uniform team composition (average 7 agents per class in each team; F-H) and diverse strategies such as camping, spreading and following are exhibited by all three agent classes. With over 80% dig sites excavated (I), the Pareto front for this scenario shows wide coverage for balancing both objectives.

This experiment shows the effectiveness of MAIM in producing teams with asymmetric agents that learn diverse synergies and specializations conducive to balancing team and conspecific objectives across a wide set of scenarios, with **(over 70% performance across the five scenarios)**.

## 5.2 Adaptation to Held-out Tasks

A crucial aspect of robust teaming is the ability to learn generalizable agent synergies and trade-offs that can be adapted to unseen changes in the environment, agent or team dynamics. We evaluate and compare MAIM's ability to adapt with several multi-objective and multiagent learning methods (section 4.2).

*5.2.1 Training on Flock.* We start by training all the baselines on Flock (section 5.1). We evaluate two variants of MAIM: 1) MAIM (S=1) which uses two mainlands (both assigned Flock); and 2) MAIM (S=3) which uses three mainlands (assigned Flock, Constrained and Dispersed). This allows MAIM (S=1) to potentially specialize like MCAA, MRL, while MAIM (S=3) learns trade-offs and agent synergies that can generalize across the three scenarios. For MAIM (S=3), we report the average team fitness on Flock.

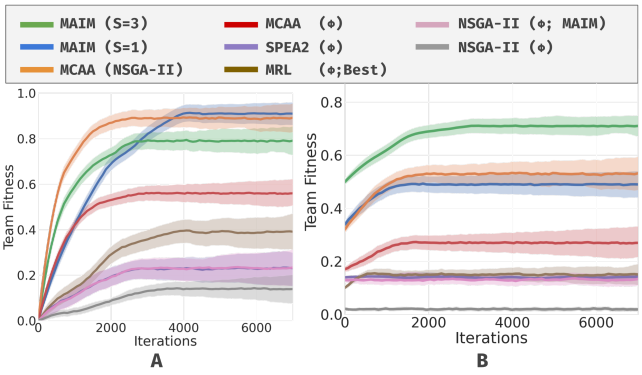


**Figure 4: Fitness of teams in Flock (A), and in the held-out Variegated task (B). In Flock, MAIM variants and MCAA have comparable team fitness with over 80% sites excavated, which is significantly higher than other baselines (A). On the held-out task, MAIM (S=3) outperforms other baselines by a wide margin (B). This highlights MAIM's ability to learn generalizable agent synergies and trade-offs that can adapt to unseen changes in the environment.**

Figure 4.A shows the performance of MAIM and the baselines trained on Flock. Because MCAA is not designed for multi-objective learning, we explicitly replace the selection mechanism in its evolutionary method with the selection mechanism of NSGA-II (2, lines 7-9). This brings MCAA closer to MAIM (S=1). Similarly, NSGA-II and SPEA2 do not account for multiagent teams, so we use the team composition learnt by MAIM (S=3) to create teams for them.

MAIM (S=1) and MCAA have comparable team fitness on Flock since both of them are equipped to learn specialized synergies and trade-offs. This is also evident form their comparable Pareto fronts (figure 5). MAIM (S=3) performs slightly worse in terms of average team fitness on Flock, but its Pareto front shows a significantly

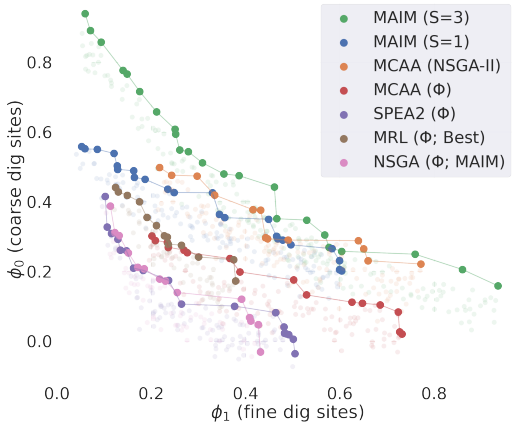


**Figure 5: Pareto fronts of teams trained with MAIM variants and the baselines. MAIM (S=3) learns a substantially higher coverage in the objective space which allows it to adapt to an unseen task without re-training (figure 4).**

higher coverage compared to the baselines. Agents trained with MRL fail to learn in this setting (performance on the best MRL island is reported) likely due to MRL's tendency to over-specialize (evident from its minimal coverage in the objective space), causing agents to maximize their conspecific utilities at the cost of team fitness [6]. Purely multi-objective methods NSGA-II and SPEA2 also perform rather poorly, partly due to their inability to learn solely with the sparse team fitness (eqn 4).

*5.2.2 Evaluation on a held-out task: Variegated.* We choose the Variegated scenario (section 5.1) as a held-out task since it offers the richest variety in dynamics amongst the five scenarios and consequently requires the agents to significantly change their inter-agent and trade-off strategies.

We allow both MAIM variants and MCAA to update their softmax distribution (which dictates the team composition), without re-training any policies on the islands or performing additional evolutionary steps on the mainlands. We also pass on the updated softmax distrbuion of MAIM (S=3) to NSGA-II and SPEA2 as before.

Agents trained with MAIM (S=3) are able to adapt their trade-offs and inter-agent relationships rapidly (figure 4.B) and maintain their average team fitness on this unseen task. The Pareto front learnt by MAIM (S=3) (figure 5) supports this as it developed the highest trade-off coverage during the initial training on Flock. The team fitness for MAIM (S=1) and MCAA reduces considerably since both methods learn to find a small subset of the Pareto front that allowed them to specialize exclusively on Flock. A similar degradation is seen for NSGA-II and SPEA2 which are unable to handle the changed dynamics of the held-out task.

This experiment highlights MAIM's ability to learn agent synergies and trade-offs that can adapt to changes in the environment.

## 5.3 Agent and Team Objective Trade-offs

Finally, we briefly explore how the team task affects inter-agent relationships and the conspecific utilities. Figure 6 shows the change in rovers' utility (left) and coverage in the objective space (right) in response to a changing environment.

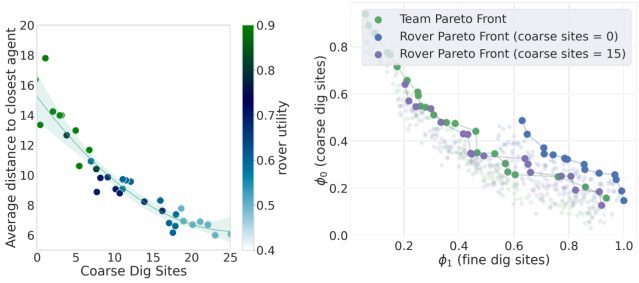


**Figure 6: [Left] As the number of coarse dig sites $K_c$ increases, rovers have to increasingly work against their preference towards fine dig sites in order to balance team objectives. This is evident from their decreasing utility and increase in inclination to team with other agent classes (decreasing inter-agent distance). [Right] Although their personal utility decreases overall, their coverage in the objective space substantially improves as they learn to balance their utility with the team objectives.**

We start by modifying Flock, by replacing all the coarse dig sites with fine sties. The conspecific utility of the rovers favours (section 4.3.1) fine sites which is fully aligned with the team objectives in the absence of coarse sites. This is evident from the initial high rover utility (figure 6, left). Although marking dig sites requires intra-agent coupling ($c$ simultaneous rovers), there is no incentive to form strong inter-agent synergies at this point. Empirically, we observe the rovers, excavators and drones spreading out in the environment to independently focus on their conspecific objectives.

Subsequently, we replace 10% of fine sites with coarse sites after every $N = 1000$ iterations. An increase in coarse sites now forces rovers to visit and mark them in order to maximize the team objective at a slight cost to their personal utility. This change is also reflected in the increased coverage of rovers in the objective space (figure 6, right) and by the increase in coupling with other agents as excavators and subsequently drones learn to follow rovers in order to balance both coarse and fine team objectives.

With its ability to concurrently balance team and individual objectives, MAIM can aid in investigating the rich evolving tapestry of agent synergies in response to the changes in other agents' utilities, the tasks and the teams in which they operate.

# 6 DISCUSSION

This work introduces MAIM, a multi-objective multiagent learning framework for the discovery of generalizable agent synergies and diverse trade-offs that foster teamwork.

MAIM leverages a Quality Diversity (QD) process that allows agents to learn diverse primitive behaviors that maximize their agent-specific utility. Concurrently, a coevolutionary algorithm evolves a population of teams (groups of agent policies) to find a Pareto front of policies that can simultaneously optimize multiple team objectives across a set of tasks. Periodic migration of policies from the highest fitness teams (on the Pareto front) to the QD process biases the diversity search process towards regions of the behavior space that yield policies conducive to good team behaviors. Likewise, the policies from the QD process are migrated to the coevolutionary process to replace the worst performing teams, thus improving the diversity in teams.

A softmax distribution governs the allocation of policies from QD to the coevolutionary process and is updated after each migration so as to maximize the cumulative agent utility across the team tasks. *The competitive pressure arising from the changing distribution of asymmetric agents across the team tasks forces agents to acquire generalizable inter-agent synergies that allow agents to exercise diverse trade-offs between agent-specific and team objectives in response to the dynamics of the task, team and agent behaviors.*

MAIM's design is primarily rooted in the known utility functions paradigm [21]. While this paradigm offers a rich set of complex problems, there is certainly value in exploring the closely related decision support paradigm, in which a priori availability of the utility functions is infeasible [22, 25]. In future work, we will explore the possibility of learning inter-agent coverage sets that allow agents to select and adapt their utilities in response to the tasks.

# REFERENCES


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

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

[3] 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

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

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[6] 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.

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

[8] 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.

[9] 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).

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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

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

[18] Jørgen Nordmoen, Kai Olav Ellefsen, and Kyrre Glette. 2018. Combining MAP-elites and incremental evolution to generate gaits for a mammalian quadruped robot. In *International Conference on the Applications of Evolutionary Computation*. Springer, 719–733.

[19] 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.

[20] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* 3 (2016), 40.

[21] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[22] Diederik M Roijers and Shimon Whiteson. 2017. Multi-Objective Decision Problems. In *Multi-Objective Decision Making*. Springer, 9–17.

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

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

[25] Luisa M Zintgraf, Diederik M Roijers, Sjoerd Linders, Catholijn M Jonker, and Ann Nowé. 2018. Ordered preference elicitation strategies for supporting multi-objective decision making. *arXiv preprint arXiv:1802.07606* (2018).

[26] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103 (2001).