MOMALAND: Benchmarking Multi-Objective Multi-Agent Reinforcement Learning

Florian Felten[a,b], Umut Ucak[a], Hicham Azmani[c], Gao Peng[d], Willem Röpke[c], Hendrik Baier[e,d], Patrick Mannion[f], Diederik M. Roijers[c,g], Jordan K. Terry[b], El-Ghazali Talbi[h,i], Grégoire Danoy[h,a], Ann Nowé[c] and Roxana Rădulescu[j,c,*]

[a]SnT, University of Luxembourg [b]Farama Foundation
[c]AI Lab, Vrije Universiteit Brussel [d]Centrum Wiskunde & Informatica
[e]Information Systems group, Eindhoven University of Technology
[f]School of Computer Science, University of Galway [g]Innovation, DII, City of Amsterdam
[h]FSTM/DCS, University of Luxembourg [i]CNRS/CRIStAL, University of Lille
[j]Intelligent Systems group, Utrecht University

Abstract. Reinforcement learning (RL) benchmarks are crucial for facilitating algorithmic progress, as well as supporting evaluation, and reproducibility in the field. This is demonstrated by the existence of numerous benchmark frameworks developed for various RL paradigms, including single-agent RL (e.g., Gymnasium), multi-agent RL (e.g., PettingZoo), and single-agent multi-objective RL (e.g., MO-Gymnasium). Multi-objective multi-agent reinforcement learning (MOMARL) is an emerging paradigm that targets complex decision-making tasks that must balance multiple conflicting objectives and coordinate the actions of various independent decision-makers. To support the advancement of the MOMARL field, we introduce MOMALAND, the first collection of standardised environments for multi-objective multi-agent reinforcement learning. MOMALAND addresses the need for comprehensive benchmarking in this emerging field, offering over 10 diverse environments that vary in the number of agents, state representations, reward structures, and utility considerations. To provide strong baselines for future research, MOMALAND also includes algorithms capable of learning policies in such settings.

1 Introduction

Many, if not most, complex problems of social relevance, such as traffic systems [17], taxation policy design [42], or infrastructure management planning [22], have both a multi-objective and a multi-agent dimension. This is because such problems often affect multiple stakeholders, who may care about different aspects of the outcome, and may have different preferences for them. As such, it is crucial to advance the field of multi-objective multi-agent decision making to enable future progress in the application of artificial intelligence (AI).

The development of standardised benchmarks is a key factor that has driven progress in various areas of AI over the years. Without standardised, publicly available benchmarks, researchers spend a lot of unnecessary time re-implementing test environments from published papers, reproducibility is made much more difficult, and results published in different papers are potentially incomparable [25, 10]. Suites of standardised benchmarks have helped to address these issues already in some fields of AI such as reinforcement learning (RL). Such benchmarks are exemplified by the seminal Gymnasium library [40] for single-objective single-agent RL, the PettingZoo library [37] for multi-agent RL (MARL), and MO-Gymnasium [2] for multi-objective RL (MORL). Yet, there is no existing library dedicated to multi-objective multi-agent reinforcement learning (MOMARL).

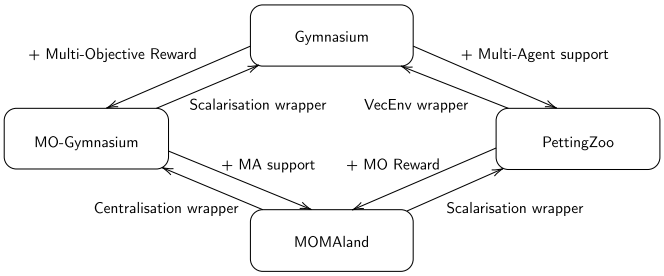

Figure 1: Overview of the libraries related to MOMAland within the Farama Foundation.

Targeting the aforementioned gap, we present MOMALAND, the first publicly available set of MOMARL benchmarks under standardised APIs. MOMALAND is constructed following the standards of the Farama Foundation ecosystem (Figure 1) and it currently offers over 10 configurable environments encompassing diverse MOMARL research settings. By embracing open-source principles and inviting contributions, we anticipate that MOMALAND will evolve in tandem with research trends and host new environments in the future.

Additionally, MOMALAND includes utilities and learning algorithms intended to establish baselines for future research in MOMARL. The utilities allow for the application of existing MORL and MARL solving methods through centralisation or scalarisation strategies. Importantly, while the provided baselines can find solutions for certain MOMARL settings, MOMALAND also features challenges with no known solution concept. Addressing these challenges requires tackling open research questions before deriving appropriate solving methods. Having set this framework, we strongly encourage contributing new work in MOMARL to the MOMALAND baselines.

* Corresponding Author. Email: r.t.radulescu@uu.nl.

2 Related Work

With millions of downloads, Gymnasium [40] (formerly known as OpenAI Gym [7]) has become the standard open source library for RL research. The varied collection of versioned environments with a standardized API allows researchers to evaluate the performance of their contributions with few code changes, and ensures valid comparisons to state-of-the-art algorithms.

However, Gymnasium is tailored for single-agent, single-objective MDPs, and does not offer support for more complex domains involving multiple agents or objectives. Hence, it has been extended in various ways, such as PettingZoo [37] or OpenSpiel [21] for MARL and MO-Gymnasium [2] for MORL.

Demonstrating the rising interest in settings involving multiple agents and objectives, some initial MOMARL benchmarks were proposed by Ajridi et al. [1] and Geng et al. [12]. Additionally, Röpke [35] introduced Ramo, a framework offering a collection of algorithms and utilities for solving multi-objective normal-form games which are a particular model studied in MOMARL. However, there is currently no widely adopted library providing reliable and maintained implementations of general MOMARL environments [18], and this is precisely the gap targeted by MOMALAND.

3 Multi-Objective Multi-Agent Reinforcement Learning

The most general framework for modelling multi-objective multi-agent decision-making settings is the multi-objective partially observable stochastic game (MOPOSG). MOPOSGs extend Markov decision processes [27] to both multiple agents and multiple objectives, under the most general setting in which agents do not observe the full state of the environment [28].

Definition 1 (Multi-objective partially observable stochastic game)**.** *A multi-objective partially observable stochastic game is a tuple* $M = (\mathcal{S}, \mathcal{A}, T, \boldsymbol{R}, \Omega, \mathcal{O})$*, with* $n \geq 2$ *agents and* $d \geq 2$ *objectives, where:*

- $\mathcal{S}$ *is the state space;*
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ *is the set of joint actions,* $\mathcal{A}_i$ *is the action set of agent* i*;*
- $T \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ *represents the probabilistic transition function;*
- $\boldsymbol{R} = \boldsymbol{R}_1 \times \cdots \times \boldsymbol{R}_n$ *are the reward functions, where* $\boldsymbol{R}_i \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ *is the vectorial reward function of agent* i *for each of the* d *objectives;*
- $\Omega = \Omega_1 \times \cdots \times \Omega_n$ *is the set of joint observations,* Ω_i *is the observation set of agent* i*;*
- $\mathcal{O} \colon \mathcal{S} \times \mathcal{A} \to \Delta(\Omega)$ *is the observation function, which maps each state – joint action pair to a probability distribution over the joint observation space.*

After every timestep, each agent receives an observation according to the observation function $\mathcal{O}$, instead of directly observing the state. In this case, memory is required for agents to successfully learn in the environment [36]. A particular form of this memory occurs when agents consider the complete history of the current trajectory denoted as $h \in \mathcal{H}$ [15] (i.e., the complete trace of executed actions and received observations).

By making additional assumptions on the MOPOSG model, regarding observability, the structure of the reward function, or whether the problem is sequential or not, we can derive a subset of models such as the multi-objective stochastic game (MOSG), multi-objective decentralised partially observable Markov decision process (MODec-POMDP), multi-objective Bayesian game (MOBG), multi-objective cooperative Bayesian game (MOCBG), multi-objective multi-agent Markov decision process (MOMMDP), multi-objective normal form game (MONFG), or multi-objective multi-agent multi-armed bandit (MOMAMAB), as illustrated in Figure 2 [28].

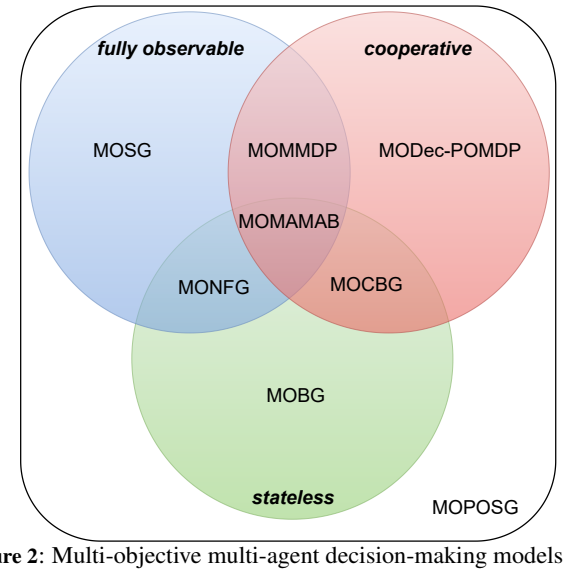

Figure 2: Multi-objective multi-agent decision-making models characterised along three axes: (i) observability; (ii) cooperativeness; (iii) statefulness [28].

In such settings, an agent behaves according to a policy $\pi_i : \mathcal{H} \times \mathcal{A}_i \to [0, 1]$, that provides a probabilistic mapping between an agent's history and its action set. In MOMARL, agents usually aim to optimise their individual expected discounted return obtained from a joint policy $\boldsymbol{\pi}$. Formally,

$$\boldsymbol{v}_i^{\boldsymbol{\pi}} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{R}_i(s_t, \boldsymbol{a}_t, s_{t+1}) \mid \boldsymbol{\pi}\right] \tag{1}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ is the joint policy of the agents acting in the environment, γ is the discount factor and $\boldsymbol{R}_i(s_t, \boldsymbol{a}_t, s_{t+1})$ is the vectorial reward obtained by agent i for the joint action $\boldsymbol{a}_t \in \mathcal{A}$ at state $s_t \in \mathcal{S}$.

Note that since an agent only directly controls its own policy π_i, this introduces subtleties not present in single-agent settings, such as non-stationarity (stemming from agents simultaneously learning in the environment) and additional credit assignment challenges (i.e., identifying the individual contribution of agents to the resulting reward signal). Moreover, as a consequence of the fact that the value function is a vector, $\boldsymbol{v}_i^{\boldsymbol{\pi}} \in \mathbb{R}^d$, it only offers a partial ordering over the policy space. Determining the optimal policy requires additional information on how agents prioritise the objectives or what their preferences over the objectives are. We can capture such a trade-off choice using a *utility function*, $u_i : \mathbb{R}^d \to \mathbb{R}$, that maps the vector to a scalar value.

In the context of multi-objective multi-agent decision-making, Rădulescu et al. [28] propose a taxonomy along the reward and utility axes. Namely, they propose to characterise settings in terms of *individual or team rewards* and *individual, team or social choice utility*. We will use the same dimensions to characterise the environments introduced by MOMALAND.

```python
from momaland.envs.multiwalker_stability import momultiwalker_stability_v0 as _env

env = _env.parallel_env(render_mode="human")
observations, infos = env.reset(seed=42)
while env.unwrapped.agents:
    actions = {agent: policies[agent](observations[agent]) for agent in
    ↪  env.unwrapped.agents}

    # vec_rewards is a dict[str, numpy array]
    observations, vec_rewards, terminations, truncations, infos = env.step(actions)

env.close()
```

LISTING 1: Parallel API usage.

```python
from momaland.envs.multiwalker_stability import momultiwalker_stability_v0 as _env

env = _env.env(render_mode="human")
env.reset(seed=42)
for agent in env.agent_iter():
    # vec_reward is a numpy array
    observation, vec_reward, termination, truncation, info = env.last()
    if termination or truncation:
        action = None
    else:
        action = policies[agent](observation)
    env.step(action)
env.close()
```

LISTING 2: AEC API usage.

4 APIs and Utilities

APIs MOMALAND extends both PettingZoo APIs by returning a vectorial reward (i.e., a NumPy [14] array) instead of a scalar for each agent.

The first API, referred to as *parallel*, enables all agents to act simultaneously, as demonstrated in Listing 1. In this mode, signals such as observations, rewards, terminations, truncations, and additional information are consolidated into dictionaries, mapping agent IDs to their respective signals (line 9). Similarly, all actions are provided simultaneously to the step function as a dictionary, mapping each agent's ID to its corresponding action (line 6).

The second API, termed *agent-environment cycle* (AEC), is suitable for turn-based scenarios, such as board games [37]. A typical usage of this API is depicted in Listing 2. In this setup, each loop provides information solely for the agent currently taking its turn (line 7).

These APIs enable modelling all our benchmarking environments and offer the advantage of aligning closely with PettingZoo's conventions, thus facilitating comprehension for MARL practitioners and reuse of existing utilities such as SuperSuit's wrappers [38]. Additionally, MOMALAND provides utilities to expose most environments through both APIs (with the exception of some board games, where support for the parallel API is deemed unnecessary).

Utilities In addition to environments and standard APIs, MOMALAND provides several utilities that help algorithm designers in creating and evaluating algorithms in the proposed environments.

These utilities are wrappers that allow modifying one aspect of the environment, such as normalising observations. MOMALAND environments are already compatible with PettingZoo and SuperSuit wrappers out of the box, as long as they do not alter the reward vectors. This allows relying on stable implementations and avoiding code duplication. However, MOMALAND provides wrappers dedicated to handling the vectorial rewards, as this is the main difference with PettingZoo. For instance, the *NormaliseReward(idx, agent)* wrapper facilitates the normalisation of the idx^{th} immediate reward component for a specified agent. Furthermore, the *LinearReward* wrapper enables the transformation of agent reward vectors into scalar values through a weighted sum of reward components, thereby converting multi-objective environments into single-objective ones under the standard PettingZoo API, see Figure 1. This adaptation allows for the utilisation of existing multi-agent RL algorithms to learn for a designated trade-off. Moreover, the *CentraliseAgent* wrapper compresses the multi-agent dimension into a single centralised agent, providing direct conversion to the MO-Gymnasium API [2]. This adaptation enables learning using multi-objective single-agent algorithms, such as those featured in MORL-Baselines [10].

5 Environments

MOMALAND provides environments with a diverse range of challenges to benchmark MOMARL algorithms. Table 1 shows an overview of all environments, together with a description of the salient dimensions in multi-objective multi-agent settings. Our environments cover discrete and continuous state and action spaces, stateless and stateful environments, cooperative and competitive settings, as well as fully and partially observable states. Some environments are multi-objective extensions of PettingZoo domains, others have been implemented from the current literature in MOMARL, and some are newly introduced, e.g., the CrazyRL variants. In the following, we briefly outline each environment.

Multi-Objective Beach Problem Domain (MO-BPD) The Multi-Objective Beach Problem Domain (MO-BPD) [24] is a setting with two objectives, reflecting the enjoyment of tourists (agents) on their respective beach sections in terms of crowdedness and diversity of attendees. Each beach section is characterised by a capacity and each agent is characterised by a type. These properties, together with

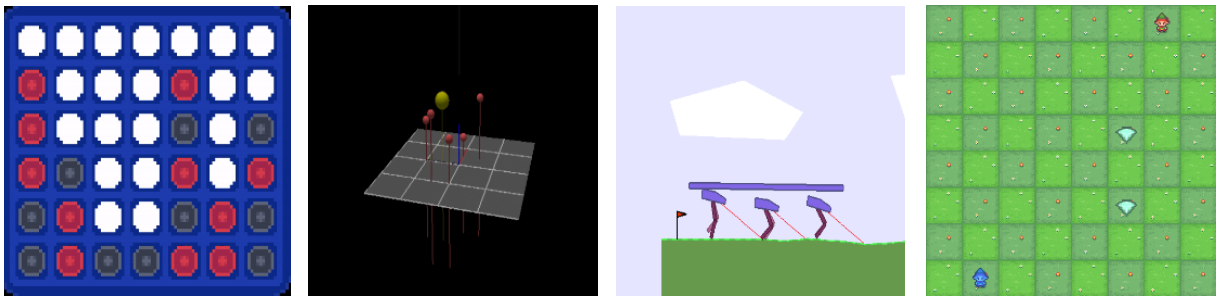

Figure 3: Visualization of some environments in MOMALAND. From left to right: MO-Connect4, CrazyRL/Surround, MO-MultiWalker-Stability, MO-ItemGathering.

Domain	# of agents	# of objectives	stochastic transitions?	full observability possible?	partial observability possible?	team rewards possible?	individual rewards possible?	discrete/continuous state (d/c)	discrete/continuous actions (d/c)
MO-BPD	2-n	2	✗*	✗	✓	✓	✓	d	d
MO-ItemGathering	2-n	2-d	✗*	✓	✗	✓	✓	d	d
MO-GemMining	2-n	2-d	✗*	-	-	✓	✗	-	d
MO-RouteChoice	2-n	2	✗*	-	-	✗	✓	-	d
MO-PistonBall	2-n	3	✗*	✗	✓	✗	✓	c	d/c
MO-MW-Stability	2-n	2	✗	✓	✓	✓	✓	c	c
CrazyRL/Surround	2-n	2	✗	✓	✗	✓	✓	c	c
CrazyRL/Escort	2-n	2	✗	✓	✗	✓	✓	c	c
CrazyRL/Catch	2-n	2	✓	✓	✗	✓	✓	c	c
MO-Breakthrough	2	1-4	✗	✓	✗	✗	✓	d	d
MO-Connect4	2	2-20	✗	✓	✗	✗	✓	d	d
MO-Ingenious	2-6	2-6	✗	✓	✓	✓	✓	d	d
MO-SameGame	1-5	2-10	✗*	✓	✗	✓	✓	d	d

Table 1: Overview of MOMALAND environments. State observability and discreteness are not specified for MO-GemMining and MO-RouteChoice as these are stateless domains. Entries marked with * denote environments that can have randomised starting states, but otherwise no stochastic transitions. Upper limits specified as n or d signal that the environment in question does not enforce an upper limit on the number of agents or objectives, respectively.

the location selected by the agents on the beach sections, determine the vectorial reward received by agents. The number of agents is configurable.

The MO-BPD domain has two reward modes: (i) *individual reward*, where each agent receives the reward signal associated with its respective beach section; and (ii) *team reward*, where the reward signal for each agent is an objective-wise sum over all the beach sections. In terms of mathematical frameworks, under the individual reward setting, the MO-BDP is a MOPOSG, while the team reward setting casts the problem as a MODec-POMDP.

MO-ItemGathering The Multi-Objective Item Gathering domain (Figure 3, rightmost picture), adapted from Källström and Heintz [20], is a multi-agent grid world, containing items of different colours. Each colour represents a different objective and the goal of the agents is to collect as many objects as possible. The environment is fully configurable in terms of grid size, number of agents, and number of objectives.

MO-ItemGathering is fully observable and has two reward modes: individual rewards (MOSG), where agents are rewarded only for their own collected items, or team rewards (MOMMDP), where agents receive a reward for any object collected by the group.

MO-GemMining In Multi-Objective Gem Mining, extending Gem Mining / Mining Day [5] to multiple objectives, a number of villages (agents) send workers to extract gems from different mines. Each gem type represents a different objective. There are restrictions on which mines can be reached from each village. Furthermore, workers influence each other in their productivity. The number of different gem types, villages, and workers per village are configurable.

MO-GemMining is stateless; each action corresponds to one independent mining day. It is fully cooperative and can be modelled as a multi-objective multi-agent multi-armed bandit (MOMAMAB).

MO-RouteChoice MO-RouteChoice is a multi-objective extension of the route choice problem [39], where a number of self-interested drivers (agents) must navigate a road network. Each driver chooses a route from a source to a destination while minimising two objectives: travel time and monetary cost. Both objectives are affected by the selected routes of the other agents, as the more agents travel on the same path, the higher the associated travel time and monetary cost. The number of agents is configurable. The environment contains various road networks from the original route choice problem [31, 39], including the Braess's paradox [6] and networks inspired by real-world cities.

MO-RouteChoice is a stateless environment, thus a MONFG, where each agent chooses one of the possible routes from its source to its destination and receives an individual reward based on the joint strategy of all agents.

MO-PistonBall MO-PistonBall is based on an environment published in PettingZoo [37] where the goal is to move a ball to the edge

of the window by operating several pistons (agents). This environment supports continuous observations and both discrete and continuous actions. In the original environment, the reward function is individual per piston and computed as a linear combination of three components. Concretely, the total reward consists of a global reward proportional to the distance to the wall, a local reward for any piston that is under the ball and a per-timestep penalty. In the MOMALAND adaptation, the environment dynamics are kept unchanged, but now each reward component is returned as an individual objective. The number of agents is configurable.

This environment is a MOPOSG, where the only stochastic transition dynamics occur when determining the initial state of the ball.

MO-MW-Stability Multi-Objective Multi Walker Stability (Figure 3, third picture from the left) is another adaptation of a PettingZoo environment, originally published in Gupta et al. [13], to multi-objective settings. In this environment, multiple walker agents aim to carry a package to the right side of the screen without falling. This environment also supports continuous observations and actions. The multi-objective version of this environment includes an additional objective to keep the package as steady as possible while moving it. Naturally, achieving higher speed entails greater shaking of the package, resulting in conflicting objectives. The number of agents is configurable.

This environment is cooperative and agents only have a partial view of the global state. Hence, it is a MODec-POMDP.

CrazyRL CrazyRL (Figure 3, second picture from the left) consists of 3 novel continuous 3D environments in which drones (agents) aim to surround a potentially moving target [8]. The two objectives of the drones are to minimise their distance to the target while maximising the distance between each other. The 3 environments differ in the behaviour of the target, which can be static, move linearly, or actively try to escape the agents.

These environments are cooperative and agents can perceive the location of everyone else. Hence, they are all MOMMDPs.

MO-Breakthrough MO-Breakthrough is a multi-objective variant of the two-player, single-objective turn-based board game Breakthrough. In MO-Breakthrough there are still two agents, but up to three objectives in addition to winning: a second objective that incentivizes faster wins, a third one for capturing opponent pieces, and a fourth one for avoiding the capture of the agent's own pieces. The board size is configurable as well.

As the game is competitive and fully observable, MO-Breakthrough falls into the category of MOSGs.

MO-Connect4 MO-Connect4 is a multi-objective variant of the two-player, single-objective turn-based board game Connect 4 (Figure 3, leftmost picture). In addition to winning, MO-Connect4 extends this game with a second objective that incentivizes faster wins, and optionally one additional objective for each column of the board that incentivizes having more tokens than the opponent in that column. As the board size is configurable, so is the number of these objectives.

MO-Connect4 is competitive and fully observable and therefore a MOSG.

MO-Ingenious MO-Ingenious is a multi-objective adaptation of the zero-sum, turn-based board game Ingenious. The game's original rules support 2-4 players collecting scores in multiple colours (objectives), with the goal of winning by maximising the minimum score over all colours. In MO-Ingenious, we leave the utility wrapper up to the users and only return the vector of scores in each colour objective. The number of agents, objectives, and board size in MO-Ingenious are configurable.

MO-Ingenious has two reward modes: (i) *individual reward*, where each agent receives scores only for their own actions; and (ii) *team reward*, where all collected scores are shared by all agents. Furthermore, it can be played with (i) *partial observability* as the original game, or in a (ii) *fully observable* mode. In terms of mathematical frameworks, this environment is therefore a MOPOSG, which can be configured to become a MODec-POMDP when playing in team reward mode, a MOSG when playing in fully observable mode, or a MOMMDP when using both.

MO-SameGame MO-SameGame is a multi-objective, multi-agent variant of the single-player, single-objective turn-based puzzle game called SameGame [4]. All legal moves in the game remove a group of tokens of the same colour from the board. The original game rewards the player for each action with a number of points that is quadratic in the size of the removed group. MO-SameGame extends this to a configurable number of agents, acting alternatingly, and a configurable number of different types of colours (objectives) to be collected.

MO-SameGame has two reward modes: (i) *individual reward*, where each agent receives points only for their own actions; and (ii) *team reward*, where all collected points are shared by all agents. It is fully observable and can therefore be modelled as a MOSG in individual reward mode, or a MOMMDP when using team rewards.

6 Baselines

After introducing our collection of challenging environments and utilities, this section demonstrates typical learning results on MOMALAND environments, for a *team reward with unknown team utility* setting. This setting aims at finding the same solution concepts as single-agent multi-objective RL, i.e., a Pareto set of policies and its linked Pareto front [16].

Using decomposition We present Algorithm 1, a simple extension of the MAPPO algorithm [41] to return a Pareto set of multi-agent policies in cooperative problems. Similar to the works of Felten et al. [9, 11], it divides the multi-objective problem into a collection of single-objective problems which can then be solved by a multi-agent RL algorithm, to obtain the components the final solution set.

In this context, a scalarisation function, parameterised by weight vectors, allows performing the decomposition and targeting various areas of the objective space. The most common scalarisation function, a weighted sum, is used in this algorithm for its simplicity (through our *LinearizeReward* wrapper, line 6). Notice that the rewards of the

Algorithm 1 MOMAPPO using Decomposition

Input: Number of weight vector candidates n, stopping criterion per weight $stop$, Environment $MOMAenv$.
Output: A Pareto set of joint policies $\mathcal{P}$.

1: $\mathcal{P} = \emptyset$
2: $\mathcal{F} = \emptyset$
3: **for** $i \in \{1, \ldots, n\}$ **do**
4: $\quad \boldsymbol{w}$ = GenerateWeights($\mathcal{F}$)
5: $\quad NormEnv$ = NormalizeRewards($MOMAenv$)
6: $\quad MAEnv$ = LinearizeRewards($NormEnv$,$\boldsymbol{w}$)
7: $\quad \boldsymbol{\pi}$ = MAPPO($MAEnv$, $stop$)
8: $\quad \tilde{\boldsymbol{v}}^{\boldsymbol{\pi}}$ = EvaluatePolicy($MOMAenv$, $\boldsymbol{\pi}$)
9: $\quad$ Add $\boldsymbol{\pi}$ to $\mathcal{P}$ and $\tilde{\boldsymbol{v}}^{\boldsymbol{\pi}}$ to $\mathcal{F}$ if $\tilde{\boldsymbol{v}}^{\boldsymbol{\pi}}$ non-dominated in $\mathcal{F}$
10: **end for**
11: **return** $\mathcal{P}$

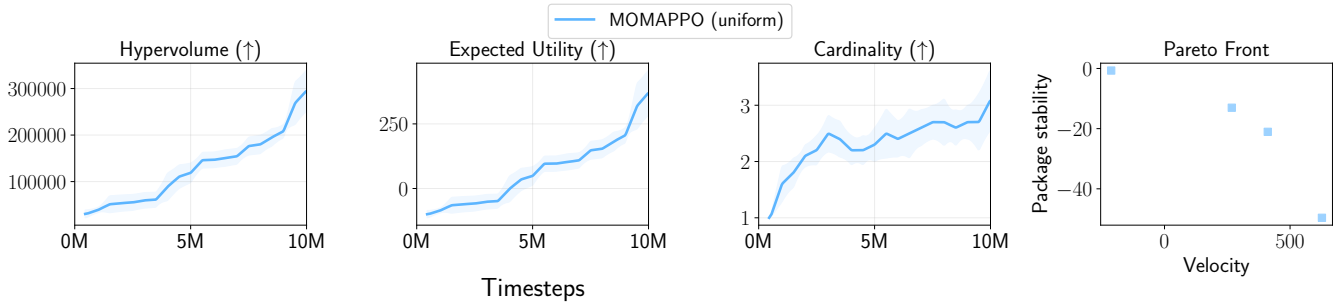

Figure 4: Average and 95% confidence intervals of multi-objective performance indicators on training results from MOMAPPO with 20 uniform weights on *mo-multiwalker-stability-v0*. The Pareto Front plot has been extracted from the run with the largest hypervolume.

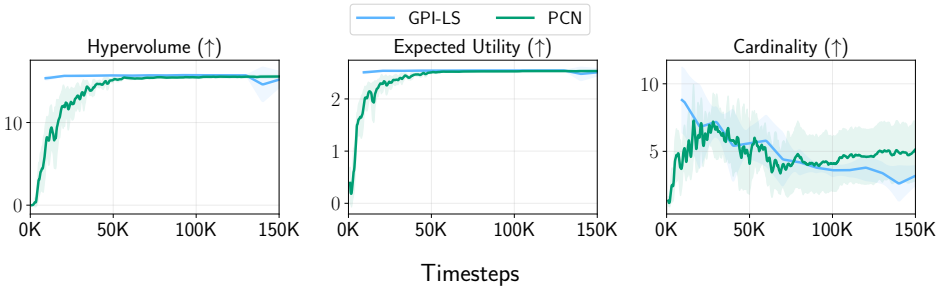

Figure 5: Average and 95% confidence intervals of multi-objective performance indicators on training results from GPI-LS and PCN on *moitem_gathering_v0*, using the centralised agent wrapper.

environment are first normalised to mitigate the difference in scale of each objective (line 5). The weight vectors are randomly generated (line 4). After training a multi-agent policy for a given trade-off using MAPPO [41], the policy is evaluated on the original environment, allowing to compute an estimate of $\boldsymbol{v}^{\boldsymbol{\pi}}$ (line 8) and to add the policy to the Pareto set of policies if it is non-dominated (line 9). Finally, the algorithm returns all non-dominated multi-agent policies (line 11).

Figure 4 illustrates the typical metrics results that can be obtained by running MOMAPPO (Algorithm 1) on a cooperative environment, *mo-multiwalker-stability-v0* in this case. For these runs, the algorithm uniformly generated 20 weight vectors to explore the objective space. The performance indicators plotted have been averaged and the 95% confidence interval is represented by the shaded area. These reflect the general performance of the algorithm over random seeds ranging between 0 and 9 included. Moreover, the PF plot gives an idea of the final result for a given run. The reference point used for hypervolume calculation is $[-300, -300]$.

The first thing to notice in the plots is that, on average, this algorithm is able to improve its PF over the training process. Indeed, all indicators improve over the training course. The PF plot reveals that 4 non-dominated policies out of 20 weight vectors have been identified. It is worth noting that this algorithm is a straightforward adaptation of MARL and MORL techniques. It can be improved by including techniques coming from existing MORL works. A thorough review of such techniques in the context of single-agent MORL is given in the work of Felten et al. [11].

Using centralisation As mentioned in Section 4, MOMALAND also provides a *CentraliseAgent* wrapper that turns a multi-agent multi-objective environment into a single-agent multi-objective environment by providing a centralised observation as well as a single vectorial reward signal. The composition method of the vectorial reward is determined by a parameter and can be either a component-wise sum or average of the individual agent rewards. This allows the direct application of methods featured in MORL-Baselines [10].

To illustrate the compatibility between MOMALAND environments using the *CentraliseAgent* wrapper and MORL-Baselines, we select two approaches, that make different assumptions regarding the environment or utility characteristics. Pareto Conditioned Networks (PCN) [32] is a multi-policy approach designed for deterministic environments. PCN will return an approximate Pareto front as a solution. On the other hand, Generalised Policy Improvement Linear Support (GPI-LS) [3] assumes the utility function is linear and will thus return the convex hull as a solution [16].

We present in Figure 5 the results obtained by GPI-LS and PCN on the *moitem_gathering_v0* environment. The experiments are run on the default map of the environment, namely an 8×8 grid, with 2 agents and 3 different object types (i.e., 3 objectives). The centralised vectorial reward signal is obtained using a component-wise addition over all agents' rewards. The number of timesteps is set to 50 and the results are averaged over 5 runs (random seeds ranging from 40 to 44), with the shaded area representing the 95% confidence interval. The reference point for the hypervolume calculation is $[0, 0, 0]$.

We observe that for this instance of the MO-ItemGathering environment, both PCN and GPI-LS show consistent learning behaviour over the runs, reaching similar performance in terms of hypervolume and expected utility. In terms of cardinality (i.e., number of solutions in the identified solution set), PCN manages to identify on average one additional solution, in comparison to GPI-LS.

7 Open Challenges

In this section we discuss notable theoretical and algorithmic challenges of MOMARL, and hope that our contribution will catalyse further research in this area.

7.1 Algorithms and Environments for MOMARL

Because it is a relatively new area, limited research has been focused on MOMARL, with only a few solving methods addressing both dimensions of the problem. Most works operate in the known utility setting, effectively relying on or adapting MARL methods, e.g. [24,

30]. A notable exception to this is MO-MIX [18], which is able to learn a Pareto set of multi-agent policies in the team reward setting. Additional research is required in general settings to establish solution concepts and develop algorithms that can identify these.

Before MOMALAND, very few environments have been identified, modelled, and made available as MOMA problems. Although we offer a preliminary set of intriguing challenges, we think this collection can be expanded and invite external contributions of new and interesting environments. For instance, the majority of the suggested environments lack a known optimal Pareto front. Knowing the optimal Pareto front would enable algorithm developers to confirm the optimality of their approaches. Another example would be contributing MOBG or MOCBG environments to the library (Figure 2). Finally, we also invite collaborations and proposals of domains based on industrial applications, especially involving environments with stochastic dynamics.

Hence, by making MOMALAND open-source and open to contributions, we hope to receive external contributions of new algorithms and environments from the research community.

7.2 Solution Concepts for MOMARL

Similar to MORL, solution concepts for MOMARL, can be defined using the two main approaches in the literature: the axiomatic approach and the utility-based approach [16]. To date, the utility-based approach has generally been the most common approach for MOMARL problems, as it allows for prior knowledge about the agents' preferences over objectives to be incorporated to simplify the problem.

When following the utility-based approach, solution concepts from traditional single-objective game theory can be extended to multi-objective settings by measuring agent incentives with respect to individual utility (rather than with respect to individual rewards/payoffs in single-objective game theory). For example, Rădulescu et al. [29] extended the well-known Nash equilibrium and correlated equilibrium solution concepts to MOMA settings using the utility-based perspective. Much of the analysis to date on solution concepts has focused on stateless single-shot settings (MONFGs), so further empirical studies are required in sequential settings. Extending existing solution concepts to MOMA settings is not trivial when following the utility-based approach, if the utility functions are non-linear. Selecting the scalarised expected return (SER) criterion in place of the expected scalarised return (ESR) [16] (or vice versa) can drastically alter the collective behaviour of the agents. For example, it has been demonstrated that it may not be possible for agents to reach a stable outcome, e.g., Nash equilibria may not exist under SER [28] or stable coalitions may not exist in coalition formation games [19]. It is also possible to have a mixture of optimisation criteria within the same system, where some agents follow SER and others follow ESR [34]. Work on such settings has been extremely limited to date and therefore further work is required to better understand the implications of mixed optimisation criteria.

Research on the axiomatic approach to MOMA problems is even less mature than the utility-based approach. The axiomatic approach may be a suitable fallback in settings where no information is available about the agents' utilities, although the space of joint policies that could be optimal is potentially much larger when no information is available about the utilities. As shown in Section 6, applying the axiomatic approach in team reward settings, where all agents receive the same reward vectors, is relatively straightforward and the problem is fully cooperative as all agent incentives are perfectly aligned. The Pareto optimal set in team reward settings simply includes all joint policies where the return vector is non-dominated. For individual reward settings (e.g., adversarial or mixed settings), Pareto optimal sets could be defined individually for each agent, as a joint policy that is Pareto optimal with respect to one agent's reward function may not necessarily be Pareto optimal for other agents. Such individual Pareto optimal sets would need to be conditioned on the behaviour of other agents in the system, so would in effect be a set of non-dominated responses to the other agents' policies [28]. When policies are deterministic with a finite number of discrete actions, the non-dominated response set for an agent would also have a finite number of policies. In settings with probabilistic policies, the non-dominated response set could potentially have an infinite number of policies.

Finally, the relationship between the axiomatic and utility-based approaches in MOMA systems is currently not well understood and merits further study. Initial work by Mannion and Rădulescu [23] in a team reward individual utility setting demonstrated that it is possible to have settings where none of the Nash equilibria are Pareto optimal, depending on the preferences of agents over objectives.

7.3 Utility Modelling and Preference Elicitation

In single-agent settings, it is possible to elicit and align preferences with respect to different trade-offs between objectives by directly interacting with the users [26, 33]. This is because it is beneficial for both the agent and the user to share such preferences openly. In multi-agent team utility settings, this would still be the case.

However, once we find ourselves in the individual utility case, the process becomes significantly harder. One may look at the problem from multiple perspectives: agents can interact and model the preferences of their users, however agents can now also potentially model their opponents' utility function, in order to gain an advantage in the strategic interactions. To the best of our knowledge, interactive MOMARL, where agents have to concurrently learn their associated user's preferences, as well as how to optimally act in the environment, has not yet been explored. Overcoming the difficulties posed by misalignment of preferences, as well as the fact that it might no longer be in the agents' best interest to share their preferences openly (on the contrary, it might even be better to actively hide this information) are still very much open challenges.

8 Conclusion

In this work, we presented MOMALAND, the first publicly available benchmark suite for MOMARL problems. Our library includes a collection of over 10 environments under two different APIs for turn-based or simultaneous actions. These environments offer a diverse set of challenges, varying in the number of agents, state and action spaces, reward structures, and utility considerations. Notably, some of these challenges have no known solution concept.

We showed how to leverage existing literature from both multi-objective RL and multi-agent RL to construct new MOMARL algorithms able to solve some of the presented challenges. These baselines, along with useful utilities, are also made available to help algorithm designers in their future research endeavours.

While the release of MOMALAND addresses one of the key challenges required to progress the field of MOMARL, many open challenges remain, as highlighted in Section 7. We hope this benchmark suite will be a valuable asset to the research community and that our work will inspire and enable future progress in the field.

Acknowledgements

This research has received funding from the project ALIGN4Energy (NWA.1389.20.251) of the research programme NWA ORC 2020 which is (partly) financed by the Dutch Research Council (NWO), and from the European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement number 101120406. The paper reflects only the authors' view and the EC is not responsible for any use that may be made of the information it contains. This work was also supported by the Fonds National de la Recherche Luxembourg (FNR), CORE program under the ADARS Project, ref. C20/IS/14762457, and by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program and by the FWO, grant number G062819N. Roxana Rădulescu was partly supported by the Research Foundation Flanders (FWO), grant number 1286223N. Willem Röpke is supported by the Research Foundation – Flanders (FWO), grant number 1197622N. We would also like to thank Lucas N. Alegre for his valuable inputs, and Manuel Goulao for helping us with the website.

References

[1] S. Ajridi, W. Röpke, A. Nowé, and R. Rădulescu. Deconstructing reinforcement learning benchmarks: Revealing the objectives. In *Proceedings of the Multi-Objective Decision Making Workshop (MODeM) at ECAI 2023*, 2023.

[2] L. N. Alegre, F. Felten, E.-G. Talbi, G. Danoy, A. Nowé, A. L. Bazzan, and B. C. da Silva. MO-Gym: A Library of Multi-Objective Reinforcement Learning Environments. In *Proceedings of the 34th Benelux Conference on Artificial Intelligence BNAIC/Benelearn*, 2022.

[3] L. N. Alegre, A. L. C. Bazzan, D. M. Roijers, and A. Nowé. Sample-Efficient Multi-Objective Learning via Generalized Policy Improvement Prioritization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[4] H. Baier and M. H. M. Winands. Nested monte-carlo tree search for online planning in large mdps. In L. D. Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, editors, *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 109–114. IOS Press, 2012. doi: 10.3233/978-1-61499-098-7-109. URL https://doi.org/10.3233/978-1-61499-098-7-109.

[5] E. Bargiacchi, T. Verstraeten, D. M. Roijers, A. Nowé, and H. Hasselt. Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In *International conference on machine learning*, pages 482–490. PMLR, 2018.

[6] D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.

[7] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym, June 2016. URL http://arxiv.org/abs/1606.01540. arXiv:1606.01540 [cs].

[8] F. Felten. *Multi-Objective Reinforcement Learning*. PhD thesis, Unilu - Université du Luxembourg [FSTM], Luxembourg, 2024. URL https://hdl.handle.net/10993/61488.

[9] F. Felten, E.-G. Talbi, and G. Danoy. MORL/D: Multi-Objective Reinforcement Learning based on Decomposition. In *International Conference in Optimization and Learning (OLA2022)*, 2022.

[10] F. Felten, L. N. Alegre, A. Nowe, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. d. Silva. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[11] F. Felten, E.-G. Talbi, and G. Danoy. Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework. *Journal of Artificial Intelligence Research*, 79:679–723, Feb. 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15702. URL https://www.jair.org/index.php/jair/article/view/15702.

[12] M. Geng, S. Pateria, B. Subagdja, and A.-H. Tan. Benchmarking marl on long horizon sequential multi-objective tasks. In *Proceedings of the 23rd Conference on Autonomous Agents and MultiAgent Systems*, AAMAS 2023. International Foundation for Autonomous Agents and Multiagent Systems, 2024.

[13] J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

[14] C. R. Harris, K. J. Millman, S. J. v. d. Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. v. Kerkwijk, M. Brett, A. Haldane, J. F. d. Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2. Publisher: Springer Science and Business Media LLC.

[15] M. Hauskrecht. Value-function approximations for partially observable markov decision processes. *J. Artif. Int. Res.*, 13(1):33–94, Aug. 2000. ISSN 1076-9757.

[16] C. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. Irissappane, P. Mannion, A. Nowe, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36, Apr. 2022. doi: 10.1007/s10458-022-09552-y.

[17] D. Houli, L. Zhiheng, and Z. Yi. Multiobjective Reinforcement Learning for Traffic Signal Control Using Vehicular Ad Hoc Network. *EURASIP Journal on Advances in Signal Processing*, 2010(1):1–7, Dec. 2010. ISSN 1687-6180. doi: 10.1155/2010/724035. URL https://asp-eurasipjournals.springeropen.com/articles/10.1155/2010/724035. Number: 1 Publisher: SpringerOpen.

[18] T. Hu, B. Luo, C. Yang, and T. Huang. MO-MIX: Multi-Objective Multi-Agent Cooperative Decision-Making With Deep Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12098–12112, Oct. 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2023.3283537. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[19] A. Igarashi and D. M. Roijers. Multi-criteria coalition formation games. In *Algorithmic Decision Theory: 5th International Conference, ADT 2017, Luxembourg, Luxembourg, October 25–27, 2017, Proceedings 5*, pages 197–213. Springer, 2017.

[20] J. Källström and F. Heintz. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS, Montreal, Canada, May 13-14, 2019*, pages 1–7, 2019.

[21] M. Lanctot, E. Lockhart, J.-B. Lespiau, V. Zambaldi, S. Upadhyay, J. Pérolat, S. Srinivasan, F. Timbers, K. Tuyls, S. Omidshafiei, D. Hennes, D. Morrill, P. Muller, T. Ewalds, R. Faulkner, J. Kramár, B. De Vylder, B. Saeta, J. Bradbury, D. Ding, S. Borgeaud, M. Lai, J. Schrittwieser, T. Anthony, E. Hughes, I. Danihelka, and J. Ryan-Davis. OpenSpiel: A Framework for Reinforcement Learning in Games, Sept. 2020. URL http://arxiv.org/abs/1908.09453. arXiv:1908.09453 [cs].

[22] P. Leroy, P. G. Morato, J. Pisane, A. Kolios, and D. Ernst. IMP-MARL: a Suite of Environments for Large-scale Infrastructure Management Planning via MARL. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=q3FJk2Nvkk.

[23] P. Mannion and R. Rădulescu. Comparing utility-based and pareto-based solution sets in multi-objective normal form games. In *2nd Multi-Objective Decision Making Workshop (MODeM 2023) @ ECAI 2023*, September 2023. URL https://modem2023.vub.ac.be/.

[24] 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.

[25] A. Patterson, S. Neumann, M. White, and A. White. Empirical Design in Reinforcement Learning, Apr. 2023. URL http://arxiv.org/abs/2304.01315. arXiv:2304.01315 [cs].

[26] M. Peschl, A. Zgonnikov, F. Oliehoek, and L. Siebert. Moral: Aligning ai with human norms through multi-objective reinforced active learning. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2, pages 1038–1046, 2022.

[27] M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[28] 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):1–52, 2020.

[29] R. Rădulescu, P. Mannion, Y. Zhang, D. M. Roijers, and A. Nowé. A utility-based analysis of equilibria in multi-objective normal-form games. *The Knowledge Engineering Review*, 35:e32, 2020.

[30] R. Rădulescu, T. Verstraeten, Y. Zhang, P. Mannion, D. M. Roijers, and A. Nowé. Opponent learning awareness and modelling in multi-objective normal form games. *Neural Computing and Applications*, pages 1–23, 2021.

[31] G. d. O. Ramos, R. Rădulescu, A. Nowé, and A. R. Tavares. Toll-based learning for minimising congestion under heterogeneous preferences. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1098–1106, 2020.

[32] M. Reymond, E. Bargiacchi, and A. Nowe. Pareto Conditioned Networks. In *The 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1110–1118. IFAAMAS, May 2022. URL https://aamas2022-conference.auckland.ac.nz.

[33] D. M. Roijers, L. M. Zintgraf, and A. Nowé. Interactive thompson sampling for multi-objective multi-armed bandits. In *Algorithmic Decision Theory: 5th International Conference, Luxembourg, Luxembourg, Proceedings 5*, pages 18–34. Springer, 2017.

[34] W. Röpke, D. M. Roijers, A. Nowé, and R. Rădulescu. On nash equilibria in normal-form games with vectorial payoffs. *Autonomous Agents and Multi-Agent Systems*, 36(2):53, Oct. 2022. ISSN 1573-7454. doi: 10.1007/s10458-022-09582-6.

[35] W. Röpke. Ramo: Rational agents with multiple objectives. https://github.com/wilrop/mo-game-theory, 2022.

[36] M. T. Spaan. Partially observable markov decision processes. In *Reinforcement learning: State-of-the-art*, pages 387–414. Springer, 2012.

[37] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[38] J. K. Terry, B. Black, and A. Hari. SuperSuit: Simple Microwrappers for Reinforcement Learning Environments, Aug. 2020. URL http://arxiv.org/abs/2008.08932. arXiv:2008.08932 [cs].

[39] L. A. Thomasini, L. N. Alegre, G. d. O. Ramos, and A. L. Bazzan. Routechoiceenv: a route choice library for multiagent reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA 2023) at AAMAS, London, UK, 2023*, 2023.

[40] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis. Gymnasium, Mar. 2023. URL https://zenodo.org/record/8127025.

[41] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=YVXaxB6L2Pl.

[42] S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. doi: 10.1126/sciadv.abk2607. URL https://www.science.org/doi/abs/10.1126/sciadv.abk2607.