Artificial Intelligence Research and Development
K. Trejo et al. (Eds.)

doi:10.3233/FAIA250587

Automated Ethical Design of Multi-Agent Reinforcement Learning Environments

Arnau MAYORAL-MACAU [a,1], Manel RODRIGUEZ-SOTO [a], Enrico MARCHESINI [b], Maite LOPEZ-SANCHEZ [c], Martí SANCHEZ-FIBLA [a], Alessandro FARINELLI [d], Juan Antonio RODRIGUEZ-AGUILAR [a]

[a] *Artificial Intelligence Research Institute, Barcelona, IIIA-CSIC*
[b] *The Massachusetts Institute of Technology (MIT)*
[c] *Universitat de Barcelona (UB)*
[d] *Università degli Studi di Verona*

Abstract. This paper introduces the Approximate Multi-Agent Ethical Embedding Process, an algorithm to ethically design reinforcement learning environments where agents learn behaviours aligned with a moral value, while pursuing their own goals. Building on Multi-Objective and Deep Reinforcement Learning, it extends a previously theory-driven method limited to small-scale problems. The new approach is tested in a scaled-up, ethically augmented version of the gathering game, demonstrating its effectiveness in managing increased complexity.

Keywords. Multi-Agent Reinforcement Learning, Value-Alignment, Multi-Objective Reinforcement Learning

1. Introduction

Autonomous artificial agents are becoming increasingly prevalent [1,2]. However, as we delegate more tasks, such as autonomous driving or healthcare, to artificial agents [3], we must also be aware of the possible risks or negative ethical effects that may arise. Thus, it is imperative to develop systems to ensure that these agents will always make decisions in alignment with human values [4].

A common approach in decision making is to let agents learn to behave using reinforcement learning (RL). Multi-agent reinforcement learning (MARL) algorithms have found application in diverse domains, exhibiting a notable capacity for acquiring proficiency in intricate tasks [5]. As a consequence, works focusing on applying RL to ensure value alignment have recently begun to appear from the fields of Machine Ethics [6] and AI Safety [7].

While the literature has largely focused on developing specific learning algorithms for value alignment problems, little attention has been given to the design of value-aligned environments, both single-agent and multi-agent, where agents learn to behave ethically regardless of the learning algorithm used. In [8], Rodriguez-Soto et al. proposed an *Ethical Embedding* algorithm that computes how to set ethical rewards neces-

[1] Corresponding Author: Arnau Mayoral-Macau (arnau.mayoral@iiia.csic.es)

sary for guaranteeing the learning of an ethically-aligned behaviour for all agents within a multi-agent system. Although their algorithm is theoretically guaranteed to succeed, it is based upon strict theoretical assumptions, such as that all agents have full observability of the whole environment or that an optimal behaviour exists for every agent independently of what the other agents are doing. These assumptions are hardly true in large and more realistic environments than those studied in [8]. Moreover, the ethical embedding algorithm requires RL algorithms with convergence properties to maintain its theoretical guarantees. To our understanding, no deep RL (DRL) algorithm preserves such guarantees. Thus, even if we found a large environment for which such theoretical assumptions held, the computational cost of computing an ethical embedding without DRL would make it unfeasible in practice.

Against this background, this work first aims to design an approximate version of the Ethical Embedding algorithm suitable for large, partially observable environments. A new method is introduced to compute the ethical embedding in multi-agent systems using DRL, allowing application in environments with more agents, larger state spaces, and partial observability. Unlike the original algorithm, this approach does not guarantee a perfectly ethical environment, leading to a second objective: defining a quality measure for value alignment.

In line with these objectives, we present our primary contribution: the *Approximate Multi-Agent Ethical Embedding Process* (AMAEEP) algorithm, extending the original *Multi-Agent Ethical Embedding Process* (MAEEP). AMAEEP transforms a multi-objective environment, where value alignment is treated independently, into a single-objective environment where agents are incentivised to behave ethically. The shift from classical RL to DRL enhances scalability but sacrifices convergence guarantees. Therefore, the resulting environment is not theoretically guaranteed to generate *perfectly* ethical agents. Nevertheless, our experiments empirically demonstrate that AMAEEP generates ethical joint policies in the Ethical Gathering Game environment, using the original map size, partial observability, and up to five agents, surpassing the reduced settings in prior work and illustrating a significant scalability improvement through DRL methods.

2. Background

This section introduces the necessary background and related work in MARL and the design of ethical environments.

The MARL literature formally defines a multi-agent environment as a *Markov game* (MG) [9]. An MG characterises an environment in which multiple agents can repeatedly act upon it to modify it, and immediately, each one receives a reward signal after each action. Formally:

Definition 1 (Markov game)**.** *A (finite) Markov game of n agents is defined as a tuple* $\mathscr{M} = \langle S, A^{i=1,\cdots,n}, R^{i=1,\cdots,n}, T, \gamma \rangle$ *containing two sets, two functions, and a constant. Here, S is a finite set of states, and* A^i *represents the set of actions available to agent i. The transition function* $T : S \times A^1 \times \cdots \times A^n \times S \to [0,1]$ *defines the probability of moving from state s to the next state* s'*, given the joint action* $a = \langle a^1, \ldots, a^n \rangle$ *of all agents. For each agent i, the reward function* $R^i : S \times A^1 \times \cdots \times A^n \times S \to \mathbb{R}$ *specifies the received reward* r^i *after applying joint action a to state s and transitioning to state* s'*. Finally,* $\gamma \in (0,1]$ *is the discount factor.*

In RL, an agent's behaviour is defined by its *policy* $\pi^i : S \to A$, mapping each state s to an action a. Each agent i seeks a policy π^i that maximises the expected discounted sum of rewards, based on its reward function R^i and discount factor γ. The combined behaviour of all agents is denoted as the joint policy $\pi = \langle \pi^1, \ldots, \pi^n \rangle$.

Since it is often impossible to find a joint policy that maximises all agents' rewards simultaneously, agents in MARL typically learn a *Nash equilibrium* (NE), a stable joint policy where no agent can unilaterally improve its outcome. Formally:

Definition 2 (Nash equilibrium)**.** *Given a Markov Game $\mathscr{M}$, a Nash equilibrium is a joint policy $\langle \pi^i_*, \pi^{-i}_* \rangle$ satisfying that for every agent i and every state $s \in S$, the policy π^i_* of agent i is a best-response against $\pi^{-i}_*(s)$, that is, it maximises the accumulation of rewards against the joint policy π^{-i}_*:*

$$V^i_{\langle \pi^i_*, \pi^{-i}_* \rangle}(s) \geq V^i_{\langle \pi^i, \pi^{-i}_* \rangle}(s), \text{ for every } \pi^i \text{ and } \forall s \in S, \tag{1}$$

where $V^i_\pi(s)$ is the expected discounted accumulation of rewards defined as $E_{\pi^i}[\sum_{t=0}^{\infty} \gamma^t r^i]$ of agent i if all agents follow the joint policy $\pi = \langle \pi^i, \pi^{-i} \rangle$.

The notion of a Nash equilibrium can be relaxed by including an $\varepsilon > 0$ in Eq. 1. When the benefit for each agent i of unilaterally modifying its policy π^i_* is at most $\varepsilon > 0$, we say that agents are in an ε-Nash equilibrium. This relaxed version of the NE is often used when computing an optimal policy is not feasible, and instead, approximations computed with algorithms without convergence guarantees are used.

Computing equilibria in an MG is a complex task that has been extensively explored by the game theory and RL literature [10]. When no specific assumptions about the game are made, employing DRL single-agent algorithms, such as Proximal Policy Optimisation [11], independently for each agent, may lead to an equilibrium [12]. However, such an approach does not have theoretical guarantees of convergence for general MGs, which might lead to suboptimal policies.

In this work, we use the term NE to refer to the joint policy obtained through a deep MARL algorithm such as Independent PPO (IPPO), while we acknowledge that the convergence to suboptimal policies of such algorithms is more accurately captured by the definition of an ε-NE.

To ensure that RL agents learn to behave ethically, we need to incorporate ethical knowledge into their environment. A typical way to aggregate ethical information is to include an ethical reward function R_e [13,8]. In this work, we focus on the approach of Rodriguez-Soto et al. in [8] because it has been shown to work for multi-agent environments.

In more detail, [8] considers a subclass of Markov games in which agents have two different reward functions R^i_0 and R^i_e. The authors formalise such a Markov game as an *Ethical Multi-Objective* Markov game:

Definition 3 (Ethical MOMG)**.** *An* Ethical Multi-Objective Markov game *is defined as a tuple $\mathscr{M} = \langle S, A^{i=1,\ldots,n}, R_0^{i=1,\ldots,n}, R_e^{i=1,\ldots,n}, T, \gamma \rangle$ such that for each agent i:*

- *R^i_0 is the original reward function of agent i, defined as reward functions in Markov games.*

- $R_e^i : \mathscr{S} \times \mathscr{A}^i \to \mathbb{R}$ *rewards performing actions ethically-aligned and punishes performing actions ethically-misaligned. To truly incentivise moral values, this reward function should be designed using principles of ethics literature, i.e utilitarianism.*

The remaining elements of $\mathscr{M}$ *are defined identically to Markov games.*

Ethical MOMGs consider alternative equilibrium concepts focusing on the different reward functions of each agent. The first of these equilibrium concepts are *ethical equilibria*, which are NE π_* with respect to the ethical reward functions R_e^i.

The second equilibrium concept of Ethical MOMGs is *best-ethical (BE) equilibrium*. Best-ethical equilibria represent joint policies in which all agents behave ethically aligned and also try to fulfil their respective individual objectives, without compromising the ethical objective. They are defined as those joint policies π_* that, among ethical equilibria, are also a NE concerning the agents' original reward function R_0^i.

The goal of the authors of [8] is to design, from a given Ethical MOMG $\mathscr{M}$, an alternative *Ethical Markov game* $\mathscr{M}_*$ that provides enough incentives to the agents to learn to behave ethically. The way of providing incentives is by designing a (single-objective) Markov game that aggregates the two reward functions of the agents $R_0^i + w_e \cdot R_e^i$ in such a way that ethical rewards R_e^i are multiplied by an ethical weight $w_e > 0$. Formally:

Definition 4 (Ethical Markov Game)**.** *Let* $\mathscr{M}$ *be an Ethical Multi-Objective Markov Game with reward functions* R_0^i, R_e^i *for each agent i. We refer to the* Ethical Markov game $\mathscr{M}_*$ *associated with* $\mathscr{M}$ *to a Markov game with reward function* $R_0^i + w_e \cdot R_e^i$ *and* $w_e > 0$*, where at least one Nash Equilibrium of* $\mathscr{M}_*$ *is a best-ethical equilibrium in* $\mathscr{M}$*.*

Rodriguez-Soto et al. provided a process in [8] to compute such an environment, called the *multi-agent ethical embedding process* (MAEEP).

MAEEP offers convergence guarantees under restrictive conditions, notably the availability of a single-agent RL algorithm that reliably finds the best response in single-agent ($n = 1$) Markov games, or *Markov Decision Processes*. Algorithms with such guarantees, like *Q-Learning* [14], enable a reliable ethical environment design. However, these algorithms struggle to scale, limiting MAEEP's applicability to larger and more realistic environments.

3. Approximate Ethical Embedding

This section introduces the *Approximate Multi-Agent Ethical Embedding Process* (AMAEEP), which designs environments that promote ethical behaviour by guiding agents towards approximate best-ethical equilibria.

The process transforms a MOMG (multi-objective Markov game) into a single-objective MG by combining individual and ethical rewards through scalarisation. This is done using the *minimal* ethical weight that still ensures ethical behaviour. The minimal weight is sought for three reasons: (1) to reduce deployment costs linked to reward shaping, (2) to align with AI Safety principles by limiting design impact [15,16], and (3) to avoid learning instability effects such as exploding gradients in DRL algorithms that may cause agents to ignore their individual goals. The original MAEEP computes

the exact minimum ethical weight required to incentivise ethical behaviour in an ethical MG. However, this necessitates the exact computation of NE, which is impractical for large state spaces. In this work, we relax this requirement by using approximations of NE through DRL algorithms that may converge to suboptimal policies but are capable of learning in large state-action spaces. As a result, the computed ethical weight is an approximation rather than the exact minimum, and the environment lacks theoretical guarantees for ensuring ethical behaviour. Despite this, experiments show that DRL converges to policies that effectively produce environments that are empirically ethical.[2]

The remainder of this section explains our approximate multi-agent ethical embedding process (AMAEEP) for designing ethical MGs. This process consists of three steps:

1. **Reference policy computation.** We compute a so-called *reference joint policy* π_r, wherein all agents behave ethically. The computation is performed by applying any algorithm to compute NE.
2. **Ethical weight computation.** We propose an iterative algorithm to find an approximation of the minimal ethical weight w_e for which the reference policy π_r is also a Nash equilibrium in an ethical MG $\mathscr{M}_*$ with associated ethical weight w_e.
3. **Build approximately an ethical environment.** We build the ethical MG $\mathscr{M}_*$ using the ethical weight w_e to scalarise and embed into a single reward function both reward functions of the original environment.

3.1. Reference policy computation

The initial phase of the AMAEEP involves identifying a best-ethical NE in the ethical MOMG $\mathscr{M}$ that will serve as the *reference policy*. Our way of computing this reference policy is by computing an NE in an auxiliary ethical Markov game, which we call a *strong* ethical Markov game $\mathscr{M}_s$. In a strong ethical MG, its associated ethical weight w_e is large enough $w_e >> 1$ to incentivise agents to always prioritise the ethical objective over the individual objective (without completely disregarding it). Thus, agents will behave ethically for any NE of a strong ethical MG. Formally:

Definition 5 (Strong ethical Markov Game)**.** *Let $\mathscr{M}$ be an Ethical Multi-Objective Markov Game with reward functions R_0^i, R_e^i for each agent i. We define a* strong ethical Markov game *$\mathscr{M}_s$ associated with $\mathscr{M}$ as a Markov game with reward function $R_0^i + w_s \cdot R_e^i$ with weight vector $w_s >> 1$ significantly larger than 1 (assuming R_0^i and R_e^i are normalised or in a similar scale), such that every Nash equilibrium in $\mathscr{M}_s$ is also a best-ethical equilibrium in $\mathscr{M}$.*

Although a strong ethical Markov game $\mathscr{M}_s$ incentivises agents to learn to behave ethically, for the three reasons exposed at the beginning of Section 3, we cannot consider $\mathscr{M}_s$ as the final environment where agents will learn to behave. At this point, we know a best-ethical NE π_r where agents behave ethically.

To finish this Subsection, we highlight the inputs of this step of the AMAEEP: an ethical MOMG $\mathscr{M}$, a weight w_s large enough, and any algorithm to compute NE *SolveMG*. If SolveMG is guaranteed to find a Nash equilibrium, the obtained reference policy will be an exact best-ethical equilibrium. Otherwise, the obtained reference policy will be an approximate best-ethical equilibrium.

[2]Even without guarantees, agents do learn ethical policies.

3.2. Minimum weight computation

After obtaining the reference policy π_r, the second step of the AMAEEP consists of finding a weight $w_e \in (0, w_s]$ that makes π_r the NE of a scalarised environment built with $R = R_0 + w_e \cdot R_e$. With such an ethical weight w_e, we will be able to design a (single-objective) ethical MG wherein at least one Nash equilibrium (the reference joint policy) is an NE.

Our ethical weight computation algorithm can be found in Algorithm 1. Our algorithm considers as input: an ethical MOMG $\mathscr{M}$, the reference joint policy π_r, a small positive number $\delta > 0$, and any algorithm to compute equilibria in a Markov game *SolveMG*.

Our ethical weight computation algorithm works as follows. First, we know that w_e is greater than 0 and smaller than or equal to w_s (because we already know that π_r is an approximate NE for the ethical weight w_s). Thus, the ethical weight w_e we seek belongs to the interval $[0, w_s]$. To obtain such weight, we iteratively select specific points w'_e of the interval $w'_e \in [0, w_s]$ following a heuristic. For each weight w'_e, we build an associated MG $\mathscr{M}_{w'_e}$. Thereafter, we compute a NE ρ within such environment $\mathscr{M}_{w'_e}$. If, for a given ethical weight w_e, the computed equilibrium π is identical to the reference policy π_r, our algorithm finishes and returns w_e as the minimal weight.

The ethical weight computation begins by computing a NE for weight $w'_e = 0$ (lines 1-3 of Algorithm 1). That is, we run the *SolveMG* algorithm on an MG with reward function $R^i_0 + 0 \cdot R^i_e$.

The algorithm finishes if the resulting NE obtains the same returns as π_r (line 4 of Algorithm 1). Otherwise, the algorithm proceeds if a different equilibrium $\pi \neq \pi_r$ is obtained. Figure 1 illustrates an example environment in which, for a given agent, the reference policy (depicted in green) and the policy associated with $w'_e = 0$ have different scalarised returns. If these two policies differ for a single agent, the algorithm needs to compute a different candidate weight.

The algorithm continues by heuristically selecting a new candidate weight w'_e inside the interval $(0, w_s]$. This new candidate weight w'_e is the point at which, for every agent i, the scalarised value of the reference policy π_r is at least as high as the value of the equilibrium π of environment $\mathscr{M}_{w_e}$ (lines 5-8 of Algorithm 1):

$$V^i_{0\,\langle \pi^i_r, \pi^{-i}_r \rangle}(s) + w'_e \cdot V^i_{e\,\langle \pi^i_r, \pi^{-i}_r \rangle}(s) \geq V^i_{0\,\langle \pi^i, \pi'^{-i} \rangle}(s) + w'_e \cdot V^i_{e\,\langle \pi^i, \pi'^{-i} \rangle}(s), \forall \text{agents } i. \tag{2}$$

Notice that such a new weight w'_e is precisely the point at which the scalarised values of π^i_r and π^i intersect for all agents i. For instance, back to Figure 1 example, assuming there is only one agent, the new candidate ethical weight w'_e is selected by comparing the point at which the blue line and the green line intersect. In this case, it is the point $w'_e = 1.59$.

Consequently, our algorithm proceeds by computing a NE for the $w'_e + \delta$ (line 9 of Algorithm 1). Recall that, for the found w'_e, both the ethical reference policy π_r and the equilibrium ρ might obtain the same scalarised value. This $\delta > 0$ is a small number to guarantee that π_r is prioritised over ρ.

Again, we build the Markov game $\mathscr{M}_{w'_e}$ associated with the new weight w'_e (line 10 of Algorithm 1), and compute an equilibrium for $\mathscr{M}_{w'_e}$ using SolveMG (line 11 of Algorithm 1). If SolveMG finds π_r, the algorithm finishes and returns the found weight

Algorithm 1 Minimum Weight Computation

Input: Ethical MOMG $\mathscr{M}$, SolveMG, δ, π_r,
1: Set the ethical weight $w_e \leftarrow 0$.
2: Set $\mathscr{M}_{w_e}$ a single-objective Markov game associated to ethical weight w_e.
3: $\rho \leftarrow SolveMG(\mathscr{M}_{w_e})$.
4: **while** $\rho \neq \pi_r$ **do**
5: **for** every agent i **do**
6: Set $w'_e \leftarrow \frac{V_0^{\rho^i} - V_0^{\pi_r^i}}{V_E^{\pi_r^i} - V_E^{\rho^i}}$.
7: Set $w_e \leftarrow \max(w_e, w'_e)$.
8: **end for**
9: Set the ethical weight $w_e \leftarrow w_e + \delta$
10: Set $\mathscr{M}_{w_e}$ a single-objective Markov game associated to ethical weight w_e.
11: $\rho \leftarrow SolveMG(\mathscr{M}_{w_e})$.
12: **end while**
13: **return** ethical weight $w_e \leftarrow w_e + \delta$.

(line 12 of Algorithm 1). Otherwise, we compute a new ethical weight again by applying Eq. 2 and repeat until convergence.

To guarantee that the algorithm always converges, the ethical weight must increase at every iteration. To guarantee that, we set the following ethical weight as the maximum among $w'_e + \delta$ and $w_e + \delta$.

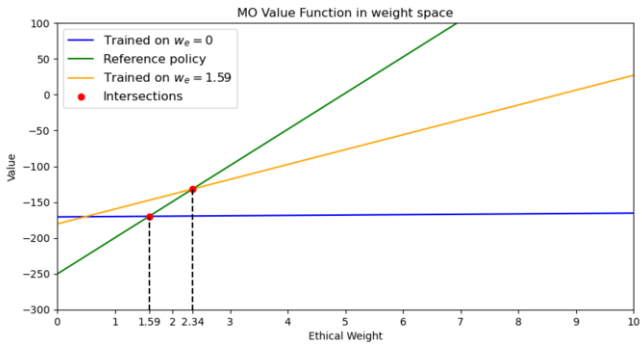

Figure 1. Representation in weight space of the scalarised values that the three policies of Table 1 obtain when scalarising their respective value vectors with an ethical weight on the weight interval [0, 10].

Training w	V_0	V_e
w_s	-250.134	20.1
0	-170.557	0.5257
1.6	-180.3455	20.72

Table 1. MO values obtained by agent i trained with different weights.

4. Experiments and results

Our experimental evaluation aims to experimentally validate our approximate multi-agent ethical embedding process with a Markov game from the literature, the *Ethical Gathering game* [17,8]. In particular, we evaluated the degree of ethical alignment of the learnt policies of the agents in the environment designed by MAEEP with two metrics:

1. For each agent, we compared their accumulation of ethical returns V_e^i with respect to the reference policy applied by our AMAEEP.
2. For each agent, we registered the number of unethical actions that they performed (i.e., actions that provide a negative ethical reward $r_e^i < 0$).

Environment description All experiments were conducted in an extended version of the *Ethical Gathering Game* (EGG) [8], a grid world where agents collect apples to survive. The EGG introduces inequality by assigning different gathering efficiencies to agents, making some more likely to survive. To promote beneficence, a donation box mechanism is added, allowing agents with surplus apples to donate and others to retrieve them.

While [8] worked in an EGG with a grid-map of 3×4, thus limiting the state space, we performed our experiments in the original Gathering Game [17] grid map (16×32). The size of our state space makes the original Ethical Embedding impractical. In addition to the augmented map size, we included 5 agents in the environment, in contrast to the two agents of [8]. Our environment has the following parameters: survival threshold $thd = 15$, donation box capacity $dbc = 10$, and partial observability of a 9×9 area. Agents can move up, down, right and left; and donate, or retrieve apples from the donation box. Agents must step on apples to collect them. Efficient agents always succeed in gathering apples while inefficient ones often fail (the apple gets lost, and they do not receive it). This results in the efficient agents achieving survival in most episodes, while the inefficient agents usually die. In our environment, only agents $i = 3$ and $i = 5$ are efficient.

Environment rewards In the EGG, agents pursue both an individual goal (maximising apple collection) and an ethical goal (supporting beneficence via donations), each defined by a distinct reward function:

Individual reward function R_0^i : Agents receive $R_0^i = -1$ per time-step until reaching their survival threshold. Gathering an apple (from the ground or donation box) grants $+1$, while donating an apple incurs a penalty of -1.

Ethical reward function R_e^i : Donating an apple after reaching survival threshold yields a reward of $R_e^i = 0.7$, while unjustified withdrawals from the donation box are penalised with $R_e^i = -1$.

4.1. Algorithm architecture

We have used an Independent PPO [12] architecture as the Markov game solver with three hidden layers of 256 units each for both the actor and the critic neural networks. To select the hyperparameters of IPPO, we applied *Optuna* [18], a hyperparameter optimiser. Specifically, we used it to set each agent's learning rate (for actors and critics) and global entropy annealing parameters. We set IPPO to do 80000 episodes of 500 time steps for all the training instances done on the experiments. Updating parameters every five episodes.

4.2. Applying the AMAEEP

Here we detail the steps of applying our ethical embedding process for our EGG.

Reference Policy Computation. The initial step in executing AMAEEP involves computing a reference policy by learning an approximate equilibrium within a *strong* ethical MG, denoted as $\mathscr{M}_s$. For this experiment, we selected $w_s = 10$ to construct $\mathscr{M}_s$, thus prioritising the ethical objective tenfold over the individual objective. Table 2 (row

Experiment	Agent 1 (ineff.)		Agent 2 (ineff.)		Agent 3 (efficient)		Agent 4 (ineff.)		Agent 5 (efficient)		Global statistics	
	V_0^1	V_e^1	V_0^2	V_e^2	V_0^3	V_e^3	V_0^4	V_e^4	V_0^5	V_e^5	Survival Rate	Full DB
large $w_e = w_s$	−319.85	0.47	−335.38	0.00	−137.98	20.92	−265.34	0.00	−164.65	15.33	100%	96%
large $w_e = 0$	−498.88	0.00	−499.51	0.00	−92.82	−0.53	−498.55	0.00	−125.33	−0.28	0%	0%
large $w_e = 2.6$	−294.13	0.53	−323.51	0.00	−124.56	20.93	−261.98	0.00	−138.02	15.95	100%	95%

Table 2. Individual returns V_0^i and ethical returns V_e^i obtained by each agent during the different steps of our AMAEEP in both the medium and large configurations and their ethical weight w_e. The two last columns show the percentage of simulations where *all* agents survive and the percentage of simulations where the donation box is full by the end of the simulation.

1) shows that, in our instance of the EGG environment, policies trained in $\mathscr{M}_s$ result in significantly higher ethical returns for efficient agents compared to inefficient agents. We can also see how the percentages regarding the survival of all agents and donation box filling are high. Furthermore, there are 0 unethical actions. This suggests that the learning algorithm effectively computed an approximation of the best-ethical equilibrium.

Minimum Weight Computation The subsequent step involves identifying the near-minimum ethical weight. Initially, it is necessary to determine the NE for the environment when the ethical weight is 0. Row 2 in Table 2 shows that the values for the individual objective V_0 for efficient agents are high, whereas those for inefficient agents are significantly low. Given that the ethical weight is zero, no agent receives a positive ethical return, as ethical actions have not been rewarded during training. Additionally, we observe there is no simulation in which all the agents survive, nor the donation box ends up full at the end of the simulation. We refer to this kind of policy as *unethical* policy.

To find the next candidate weight w_e', we apply equation 2 with the value vectors corresponding to scalarised environments $w_e = 0$ and $w_e = w_s$ (rows 1 and 2). To maintain brevity, we do not describe the computations for all agents; we focus only on agent $i = 5$ (columns 10, 11). The intersection (Eq. 2) of $-164 + 15.33w$ and $-125.33 - 0.28w$ is in $w = 2.51$ which will be or the next candidate weight $w_e' = 2.51$

To clarify, as stated in subsection 3.2, we select the maximum weight from the outcomes of intersecting the two policies for each agent. Additionally, we add a small δ to select a weight to the right of the intersection. Then our next candidate weight is $w_e' = 2.51 + \delta = 2.6$.

We can again build a Markov game for the obtained ethical weight w_e' and compute an equilibrium. In Table 2, the third row shows the value vector obtained in the new approximate equilibria found for $w_e' = 2.6$. We observed almost no difference in the ethical returns of the policies of efficient agents between having trained applied weights w_s or w_e'. Additionally, as the reference policy, the approximate equilibrium found for the new ethical weights commits exactly 0 unethical actions in 1000 simulations of 500 time steps. Overall, we consider that the algorithm has converged on iteration one. Thus, AMAEEP has found the best-ethical equilibrium with ethical weight set to 2.6. With such weight, we can build the final ethical MG, which the algorithm will return.

Following the procedure depicted in subsection 4.2, we have designed an ethical MG. Note that after the AMAEEP is done, there is no need to compute the NE on the resulting environment, as we obtained it as the last step of the process.

4.3. Results

In light of our experiments, we consider that the ethical design of an EGG environment using our new AMAEEP is possible. Thus, we fulfilled our primary research objective. Furthermore, to cope with the second objective we provide distinct metrics to asses that our results are indeed ethical. The similarity of the value vectors and in the EGG specific

metrics (survival and filling of the donation box) empirically proves that even when using approximations of equilibria that are as ethical as the reference policy computed.

5. Conclusions and future work

Based on the MORL literature, we tackle the open problem of building an *ethical* environment for large multi-agent systems wherein all agents in the system learn to behave ethically while pursuing their individual objectives. We call our method Approximate Multi-Agent Ethical Embedding Process (AMAEEP), and we empirically evaluated it in an ethical extension of the gathering game where agents needed to consider the moral value of beneficence. As future work, we plan to develop methods for aligning a multi-agent system with multiple moral values.

References

[1] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583-9.
[2] Wurman PR, Barrett S, Kawamoto K, MacGlashan J, Subramanian K, Walsh TJ, et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. Nature. 2022;602(7896):223-8.
[3] Boada JP, Maestre BR, Genís CT. The ethical issues of social assistive robotics: A critical literature review. Technology in Society. 2021;67:101726.
[4] Gabriel I. Artificial Intelligence, Values, and Alignment. Minds and Machines. 2020 09;30:411-37.
[5] Casas-Roma J, Conesa J. Towards the design of ethically-aware pedagogical conversational agents. In: Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 15th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2020) 15. Springer; 2021. p. 188-98.
[6] Yu H, Shen Z, Miao C, Leung C, Lesser VR, Yang Q. Building Ethics into Artificial Intelligence. In: IJCAI; 2018. p. 5527–5533.
[7] Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D. Concrete problems in AI safety. arXiv preprint arXiv:160606565. 2016.
[8] Rodriguez-Soto M, Lopez-Sanchez M, Rodriguez-Aguilar JA. Multi-objective reinforcement learning for designing ethical multi-agent environments. Neural Computing and Applications. 2023:1-26.
[9] Albrecht SV, Christianos F, Schäfer L. Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. MIT Press; 2024. Available from: https://www.marl-book.com.
[10] Papadimitriou CH, Roughgarden T. Computing equilibria in multi-player games. In: SODA. vol. 5; 2005. p. 82-91.
[11] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal Policy Optimization Algorithms; 2017.
[12] de Witt CS, Gupta T, Makoviichuk D, Makoviychuk V, Torr PHS, Sun M, et al.. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?. arXiv; 2020.
[13] Wu YH, Lin SD. A low-cost ethics shaping approach for designing reinforcement learning agents. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32; 2018. .
[14] Watkin C. Q-learning, technical note. Mach Learn. 1992;8:279-92.
[15] Hernández-Orallo J, Martínez-Plumed F, Avin S, Heigeartaigh SO. Surveying Safety-relevant AI characteristics. In: AAAI workshop on artificial intelligence safety (SafeAI 2019). CEUR Workshop Proceedings; 2019. p. 1-9. Available from: https://riunet.upv.es/handle/10251/146561.
[16] Hendrycks D, Carlini N, Schulman J, Steinhardt J. Unsolved problems in ml safety. arXiv preprint arXiv:210913916. 2021.
[17] Leibo JZ, Zambaldi VF, Lanctot M, Marecki J, Graepel T. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. CoRR. 2017;abs/1702.03037. Available from: http://arxiv.org/abs/1702.03037.
[18] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining; 2019. p. 2623-31.