Learning to identify and settle dilemmas through contextual user preferences

Rémy Chaput
Univ Lyon, UCBL, CNRS, INSA Lyon
LIRIS, UMR5205
Villeurbanne, France
remy.chaput@univ-lyon1.fr

Laetitia Matignon
Univ Lyon, UCBL, CNRS, INSA Lyon
LIRIS, UMR5205
Villeurbanne, France
laetitia.matignon@univ-lyon1.fr

Mathieu Guillermin
CONFLUENCE: Sciences et Humanités
research unit (EA 1598)
Lyon Catholic University
Lyon, France
mguillermin@univ-catholyon.fr

***Abstract*—Artificial Intelligence systems have a significant impact on human lives. Machine Ethics tries to align these systems with human values, by integrating "ethical considerations". However, most approaches consider a single objective, and thus cannot accommodate different, contextual human preferences. Multi-Objective Reinforcement Learning algorithms account for various preferences, but they often are not intelligible nor contextual (e.g., weighted preferences). Our novel approach identifies dilemmas, presents them to users, and learns to settle them, based on intelligible and contextualized preferences over actions. We intend to maximize understandability and opportunities for user-system co-construction by showing dilemmas, and triggering interactions, thus empowering users. The block-based architecture enables leveraging simple mechanisms that can be updated and improved. Validation on a Smart Grid use-case shows that our algorithm finds actions for various trade-offs, and quickly learns to settle dilemmas, reducing the cognitive load on users.**

***Index Terms*—Machine Ethics, Multi-Objective Reinforcement Learning, Moral Dilemmas, Human Preferences**

I. Introduction

With the recent progresses of Artificial Intelligence (AI), and seeing that "algorithms initially developed in the lab are increasingly being improved and deployed in society" [1], there is a crucial and pressing matter of ensuring that AI systems are aligned with (moral) values important to humans. These systems, by interacting with humans and being immersed in our societies, have an impact on our lives, making them *ethical impact agents* [2]. Thus, the designers' goal is to make these agents capable of acting from ethical considerations (*explicit ethical agents*). This is done in Machine Ethics [3] by implementing specific capabilities into them.

However, we argue that some situations, which we call *dilemmas*, cannot be "autonomously" settled by machines only, at least not how humans would like them to be settled. In these dilemmas, several moral values are in conflict, and no single decision allows satisfying all of them at the same time: each choice will lead to regret. The AI system could simply take a decision randomly, or selecting the decision that minimizes the sum of harm; but it is unlikely that such a decision process would meet the humans (regulators, users, stakeholders, etc.) expectations. Instead, we propose to leverage human preferences to settle such dilemmas.

We contend that such preferences should be integrated into the systems in an *intelligible* and *actionable* way for humans. Many algorithms, especially within the Multi-Objective Reinforcement Learning (MORL) field, find optimal solutions with vectors of weights as preferences, which is not ideal for humans. Preferences may depend upon contexts [4], e.g., persons who cannot stand the cold might prefer their well-being to ecology during winter. Defining these preferences *a priori* might reveal an unfeasible (or at least, daunting) task for humans, who would thus have to foresee the potential contexts and describe their preferences for each of them.

We propose a method that places the human back into the loop and makes explicit the notions of dilemmas and acceptable actions. Beyond finding optimal solutions by hiding away those dilemmas, our algorithm signals to users the situations in tension that require particular attention, while automating "simple" cases. This is in line with an *ethical companion* approach, in which AI systems and humans can learn from each other, in a co-construction loop.

Our contribution can be summarized as follows: we present the QSOM-MORL algorithm that learns to identify and settle dilemmas according to users' preferences, using a block-based architecture allowing for future improvements; QSOM-MORL is validated on a Smart Grid experiment.

II. State of the Art

We begin with the field of Machine Ethics, as it brings into light several requirements and desiderata that shaped our exploration of the vast field of MORL. Other fields, such as computational social choice, are related to our work but to a lesser extent. Thus, we do not delve into them here and refer the interested reader to [5] for details.

A. Machine Ethics

Machine Ethics [3] attempts to give AI systems capabilities to take ethical considerations into account in their decision-making processes. Although many approaches consider a single agent, some researchers argue that "Ethics is inherently a multiagent concern — an amalgam of (1) one party's concern for another and (2) a notion of justice" [6]. Traditionally,

This work was funded by the French Region Auvergne-Rhône-Alpes (Pack Ambition Recherche; Ethics.AI project); and by ANR (ANR-22-CE23-0028-01; ACCELER-AI project).

approaches in Machine Ethics are classified into *Top-Down*, *Bottom-Up*, and *Hybrid* approaches.

1) Top-Down formalizations: They implement ethical principles from moral philosophy, e.g., Kant's Categorical Imperative. One representative approach is Ethicaa [7], which considers multiple ethical principles with a priority order. Agents filter out actions evaluated as immoral by their preferred ethical principle until a single one remains. These approaches often exploit symbolic programming, have strong philosophical foundations, and leverage domain expert knowledge; but they fail to adapt to changes in the ethical considerations.

2) Bottom-up learning: They learn a new principle from experiences, either with a dataset of (ethically-imbued) examples, or through interactions with a simulator. While multi-objective approaches seem interesting for ethical considerations [8], most studies focus on a single ethical objective (for example, [9]), which does not prioritize moral values when they conflict. One of the few multi-objective works decompose the reward function into 3 components [10], and first finds the *ethical policies* that are optimal w.r.t. the positive and negative moral rewards only; then, it selects the *ethical-optimal* with the maximum task-specific rewards, among the ethical set. Policies thus cannot trade a lower reward on the moral components for a higher reward on the task component. However, this approach does not integrate human preferences nor contexts.

3) Hybrid: Hybrid approaches combine advantages of top-down formalizations and bottom-up learning. AJAR [11] combines symbolic-based reward functions with a multi-agent reinforcement learning algorithm that learns to respect the ethical considerations embedded in the rewards.

On the one hand, learning agents in AJAR use the QSOM algorithm [12] to learn the state-action pairs' interests. QSOM combines a Q-table and Q-learning like updates [13] with two self Self-Organizing Maps (SOMs) [14] to handle continuous and multidimensional states and actions. An important advantage of QSOM is that Q-values are easily available through the Q-table, contrary to, e.g., deep neural networks approaches that hide the interests within the weights of their networks. This is essential to compare Q-values, which we will leverage to identify dilemmas in our contribution.

On the other hand, reward functions in AJAR are defined by symbolic moral judging agents. Each judging agent is responsible for an objective and uses an argumentation graph to judge the behavior of learning agents w.r.t. its own moral value. Because QSOM agents are single-objective, AJAR scalarizes the multiple rewards from judging agents into a single number. Nonetheless, this is a suitable basis for our work, as it provides multiple objectives that focus on ethical considerations. Thus, our contribution builds upon the QSOM learning algorithm, and uses vectorized AJAR reward functions.

B. Multi-Objective Reinforcement Learning

The MORL field deals with optimizing several potentially conflicting objectives simultaneously. However, it is not always possible to fully satisfy all objectives at the same time, as demonstrated in the Deep Sea Treasure benchmark [15]: we have to choose between valuable items and leaving in a minimal time.

Many works propose scalarization as a solution to reduce the multi-objective problem into a single-objective one, yet some authors argue that it is not ideal [16, p.2]. We argue that the following reasons are especially important when dealing with *ethical* objectives: 1) it puts too much burden on the (AI) engineers, instead of opening choices to people (users, domain experts, philosophers, etc.); 2) the resulting behavior is harder to explain, because objectives are "blended" together; 3) it cannot handle changing preferences.

We thus focus on approaches that explicitly target multiple objectives, without scalarization; they return a set of policies, instead of a single optimal policy. Some rely on the "convex hull" [17], [18]: instead of learning a *Q-table*, they learn the *hulls* that correspond to the space of interests w.r.t. preferences. At the execution time, users specify their preferences to retrieve the corresponding interests and optimal policy from the convex hull. However, this requires the users' preferences for each objective to be a vector of weights. Yet, there is no linear relationship between the weights and the resulting policy: it can be hard to obtain what we truly desire. Thus, our contribution explicitly identifies situations of dilemmas, presents them to users, and learns to settle them, based on contextualized preferences that users explicitly specify.

Finally, few approaches consider both multi-agent and multi-objective environments [16]. Our work considers multiple agents and multiple ethical objectives, with a focus on the multi-objective part. We address the multi-agent aspect through independent learners who receive individual rewards from reward functions taking a globalized state as input.

III. The QSOM-MORL approach

This section presents our proposed QSOM-MORL approach, which builds upon the QSOM algorithm [12]. QSOM-MORL is a Multi-Reward Partially Observable Markov Decision Process (MRPOMDP) [19], a generalization of the traditional MOMDP [16]. The main features are: multiple agents that receive *partial continuous observations* about the states, *continuous* (parameterized) *actions*, and a *vectorized reward function*. The goal is to learn *ethical objectives* by leveraging, in an *intelligible* and *actionable* manner, *human-specified* and *context-specific* preferences.

Definition 1 (Ethical objective): An ethical objective is an objective that drives the learning agent towards best compliance with a given ethical consideration or moral value.

The distinction between moral value and ethical objective avoids a (potential) theoretical difficulty: it might be unfeasible to formally specify some moral values, such as "human dignity". Yet, we can identify important aspects of the moral values, e.g., "killing someone violates their human dignity", and thus to correspondingly reward or punish the learning agent, even though all corner cases might not be covered for such intricate moral values. In QSOM-MORL, we consider $m > 1$ ethical objectives, each associated to a moral value.

A. QSOM-MORL architecture

QSOM-MORL is conceived as a high-level architecture with several "building blocks"; each solves a part of the MORL problem, interacts with the others, and could be replaced individually. The main advantage is to propose a complete, working algorithm and proof-of-concept, while allowing for flexibility, and ulterior improvements. Some blocks are voluntarily simple, so as to focus on the *dilemmas* and *human preferences* aspects. Our architecture comprises (Fig. 1):

- Learning interesting actions: exploring the action space, finding actions that are "interesting" (for multiple preferences), and learning their interests. It is necessary to be able to propose these actions in dilemmas.
- Identifying dilemmas: identifying which situations truly are dilemmas, to avoid asking users each time. The system can automatically settle "simple" situations.
- Aggregating dilemmas: grouping similar dilemmas that can be settled in the same manner.
- Learning the user preferences: learning the desired preferences from the human users, for each group of aggregated dilemmas. The system must re-use the same preferences for new encountered dilemmas from the same groups.

We detail each of these blocks and how they interact with each other in the subsequent sections[1].

B. Learning interesting actions

We need to: 1) accurately know the actions' Q-values, to compare them and identify whether a situation is a dilemma; and 2) know "interesting" actions that we can propose to users when in a dilemma. This requires exploring the action space, to learn the action parameters that yield good trade-offs between the moral values, for a variety of preferences.

QSOM [12] explores the action space by noising the action parameters, and observing whether the *perturbed* action performs better than expected:

$$r_t + \gamma \operatorname*{argmax}_{j'} \mathtt{Q}(s_{t+1}, j') \overset{?}{>} \mathtt{Q}(s_t, j) \tag{1}$$

In Eq. 1, the agent in state s_t performs a slightly perturbed action j'. If the received reward r_t and the expected return in the new state s_{t+1} are higher than the (previously) stored Q-value of the unperturbed action j, then the noise is considered an improvement, and the parameters of j are updated towards its noisy version[2]. When rewards and Q-values are vectors, the formula compares two vectors, and the greater operator is no longer defined. Because we do not know users' preferences *a priori*, we must explore for multiple preferences, instead of, e.g., maximizing the average return. We cannot ask users preemptively, as weight vectors would not be intelligible, especially for lay users.

Instead, we propose to create *exploration profiles*, which combine all structures from the classic QSOM and a vector of weights to **drive the exploration of the action space towards a certain subspace of the space of objectives**. Multiple exploration profiles are represented by multiple agents that learn the action space concurrently in a *bootstrap* phase, in order to explore all subspaces, thus finding interesting actions.

Definition 2 (Exploration profile): An exploration profile $p \in \mathcal{P}$ contains the following data structures and functions:

- $\texttt{States}_p : \mathbb{O} \to \mathcal{S}$ maps observations to a discrete state identifier, based on a State-SOM. Possible states are $\mathcal{S} = [[0, \cdots, |\mathbf{U}|]]$, with $|\mathbf{U}|$ neurons.
- $\texttt{Actions}_p : \mathcal{A} \to \mathbb{A}$ maps a discrete action identifier to its parameters, based on an Action-SOM. Possible action identifiers are $\mathcal{A} = [[0, \cdots, |\mathbf{W}|]]$, with $|\mathbf{W}|$ neurons.
- $\mathtt{Q}_p : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ returns the *interests* of an action in a given state, based on a Q-table, for m objectives.
- $\vec{\rho} \in \mathbb{R}^m$ are the exploration weights, used to scalarize interests to determine whether an action is interesting.

Eq. 2 extends Eq. 1 to handle vectors of rewards $\vec{r_t}$ (one reward for each ethical objective) and interests. These vectors are scalarized through a dot product $\cdot$ with the exploration weights ρ, defined as: $\vec{x} \cdot \vec{y} = \sum_{i=1}^{m} x_i y_i$.

$$\vec{\rho} \cdot \vec{r_t} + \gamma \operatorname*{argmax}_{j'} \left(\vec{\rho} \cdot \mathtt{Q}_p(s_{t+1}, j')\right) \overset{?}{>} \vec{\rho} \cdot \mathtt{Q}_p(s_t, j) \tag{2}$$

We create $m+1$ exploration profiles. A "generalist" profile, with weights $\left[\frac{1}{m}, \cdots, \frac{1}{m}\right]$, finds actions that are good in average. The m remaining profiles are each "specialized" for a specific objective i, by using a very high weight for the i-th component, and a low non-zero weight for others. For example, profile $i = 0$ will use $\left[0.9, \frac{0.1}{m-1}, \cdots, \frac{0.1}{m-1}\right]$. Non-zero weight considers improvements in other dimensions, even when the targeted one remains unchanged, e.g., consider learned interests of $[0.8, 0.3]$ and received reward of $[0.8, 0.4]$. Because each weight is > 0, a slight improvement in other objectives is acknowledged, making the perturbed action more appealing than the learned one. Yet, due to low weights, an eventual decrease in the targeted objective cannot be offset by an increase in others.

Finally, the Bellman equation [20] is also updated to take into account the multi-objective aspect of the Q-table:

$$\forall k \in [[1, m]] : \mathtt{Q}_p^{t+1}(s_t, a_t, k) \leftarrow \alpha \left[r_{t,k} + \gamma \max_{a'} \rho \cdot \mathtt{Q}_p^{t}(s_{t+1}, a', k) \right] + (1 - \alpha) \mathtt{Q}_p^{t}(s_t, a_t, k) \tag{3}$$

C. Identifying dilemmas

Once interesting actions have been learned, we leverage them during the *execution* phase to determine whether a situation is a dilemma. All exploration profiles for a same agent profile are merged, so that all learned actions are accessible.

First, we introduce *theoretical interests*, as the interests an action would have, if its impact was "perfect" and the rewards always maximal. They serve as a comparison point, to make interests more accessible to human users. Theoretical interests $\mathtt{Q}^{th}$ are learned by replacing interests $\mathtt{Q}$ by $\mathtt{Q}^{th}$ and reward $r_{t,k}$ by the maximal reward $\hat{r}$ in Eq. 3.

[1] Source code is available at https://doi.org/10.5281/zenodo.8335307

[2] We recall that we use SOMs to discretize the spaces: actions are represented by neurons and prototype vectors in a latent space.

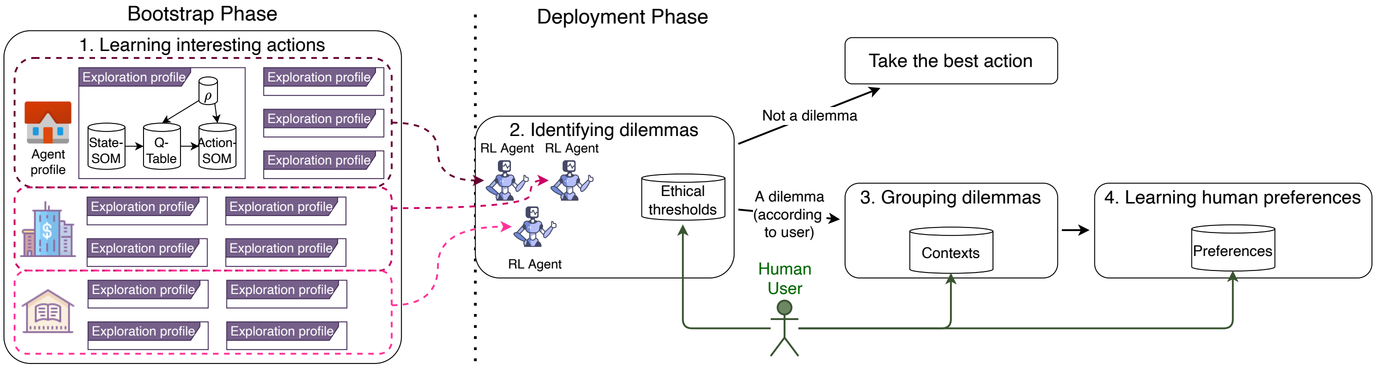

Fig. 1. Proposed architecture, comprised of several blocks; it supports heterogeneous agents, represented as *agent profiles*.

Comparing an action's interests to its theoretical interests ($\frac{\mathtt{Q}(s,a)}{\mathtt{Q}^{th}(s,a)}$) gives a *ratio* of action satisfaction (for each objective) between 0 and 1 that, we argue, is more intuitive to users. We then define the *ethical thresholds* as the users' expectations, w.r.t. ethical objectives. They are determined by users, because humans are our "source of truth" for ethics. Ethical thresholds may differ from one ethical objective to another; e.g., an ecologist might require a higher satisfaction for an ecology-related moral value. In addition, we may very well accept various "trade-offs" between moral values, e.g., "at least 60% of ecology and well-being, *or* 80% of ecology but only 50% of well-being".

Definition 3 (Ethical thresholds): An ethical threshold is a set of vectors $\zeta \in \mathbb{Z}$, where each vector represents a conjunction of m constraints (joined by *and* relations) over an action's satisfaction, m being the number of ethical objectives. The set is a disjunction (joined by *or* relations) of one or several such vectors. To simplify, we note $\zeta_{i,j}$ the j-th constraint of the i-th vector in the set. Thus, $\mathbb{Z} = \{\zeta \mid \exists n \in \mathbb{N} \ : \ \zeta \in \mathbb{R}^{m*n}\}$.

Users are not required to define several vectors; this definition supersedes "simple" cases, e.g., a single vector $(0.8 \wedge 0.5)$, or even no constraint $(0 \wedge 0)$. Its advantage is to allow for more use-cases. We can now identify *acceptable* actions.

Definition 4 (Acceptable action): An action $(p,a) \in (\mathcal{P} \times \mathcal{A})$ is deemed *acceptable*, in a situation represented by observations $\vec{o}$, if the ratio of interests over theoretical interests satisfy the ethical thresholds. Formally, $\mathtt{acceptable}(\vec{o},p,a,\zeta) \Leftrightarrow \exists i \ \forall k \in [[1,m]] \ \frac{\mathtt{Q}_p(\mathtt{States}_p(\vec{o}),a,k)}{\mathtt{Q}_p^{th}(\mathtt{States}_p(\vec{o}),a,k)} \geq \zeta_{i,k}$.

Action acceptability ultimately depends on the users, as they define ethical thresholds; this definition is key to defining whether a situation is a dilemma. We propose that, if at least one action is deemed acceptable, the situation is not treated as a dilemma, because the agent can perform this action autonomously. The user, through the ethical thresholds, has validated that this action is aligned with its preferences. Yet, if no single action is acceptable, then the situation is a *dilemma*: we cannot satisfy all objectives at the same time. Each action choice will imply a trade-off between ethical objectives.

Definition 5 (Dilemma): A situation described by observations $\vec{o}$ is in a dilemma if no proposed action (p,a) is *acceptable* w.r.t. ethical thresholds ζ. Formally: $\mathtt{dilemma}(\vec{o},\zeta) \Leftrightarrow \nexists (p,a) \in (\mathcal{P},\mathcal{A}) \ : \ \mathtt{acceptable}(\vec{o},p,a,\zeta)$.

D. Aggregating dilemmas

Some dilemmas may be similar, in the sense that they can be settled with the same action. Because users must provide their (situation-specific) preferences when dilemmas are identified, we suggest aggregating such similar dilemmas, to reduce the number of necessary user interactions.

Typically, *clustering* techniques are used to automatically determine representative clusters and assign elements to them. We prefer not to leverage such automated techniques, to avoid injecting "false" (or wrong) notions in the dilemma space: which distance metric should be used? Which threshold to accept a new dilemma? etc. These questions are often answered by fine-tuning the algorithm's hyper-parameters, however designers cannot do this *a priori*, as users may have different views on whether two given dilemmas are similar. Instead, we aggregate by asking users to define *contexts*, when a new dilemma is identified. The system then leverages contexts to automatically classify new dilemmas, or to raise the question when no context recognizes the new dilemma.

Definition 6 (Context): A *context* is a set of minimal and maximal bounds, for each dimension of the observation space $\mathbb{O} \subseteq \mathbb{R}^g$. Formally, a context c is a tuple $\langle (b_1, B_1), \cdots, (b_g, B_g) \rangle$, with g the number of dimensions of the observation space, b_k the lower bound for dimension k, and B_k the upper bound for dimension k. We say that a context *recognizes* a situation in a dilemma, represented by its observation vector $\vec{o}$, if all observations are within the context's bounds: $\mathtt{recognize}(c,\vec{o}) \Leftrightarrow \forall k \in [[1,g]] \ : \ b_k \leq o_k \leq B_k$.

E. Learning user preferences

When the system identifies a dilemma, we would like to automatically perform the action that corresponds to the user's preferences, for this context. To do so, we propose that, when a new context is created by the user, it is associated with an action choice among the proposed alternatives. Then, each time a dilemma is recognized by this context, the system automatically selects this memorized action.

To simplify the choice for the human users, we filter proposed actions based on two criteria. 1) Actions that are *Pareto-dominated* only lead to regret, compared to others. We find non-dominated actions, for which there is no other action with at least the same interests on all ethical objectives, and at least one objective with a strictly greater interest, by computing the Pareto Front (PF). 2) Actions with parameters too much similar to other actions would essentially perform the same effect. For each action, if another one exists in the PF with parameters that differ by less than a threshold (e.g., 3%) on *all* dimensions, we remove this action.

The user selects the action that best align with their preferences through a Graphical User Interface that presents the filtered alternatives, the current situation, the actions' interests and parameters. The agent memorizes the mapping $\langle c, (p, a) \rangle$ between the context and the chosen action, and automatically performs the same action (p, a) when a new dilemma is recognized by the same context c.

IV. Experiments and results

Several MORL benchmark environments exist [16], but none meet our needs: most consider only 2 objectives, and few consider continuous states and actions. Thus, we use the Smart Grid simulator of [12], where agents represent buildings and consume energy to satisfy inhabitants' comfort. States are described by a vector $\in \mathbb{R}^{11}$ of individual (e.g., agent's personal battery) and shared observations (e.g., available energy in the micro-grid). Actions are vectors of parameters $\in \mathbb{R}^6$ to handle energy from the personal battery, Smart-Grid and national grid. We use the $m = 4$ ethical objectives from AJAR [11]: *well-being*, *affordability*, *equity*, and *environmental sustainability*.

A. Agents learn various actions

Because our resulting policies depend on the human preferences, we cannot directly use the Pareto Front as validation, as many MORL approaches do with the hypervolume or sparsity. Yet, to provide an idea of the learning quality of our *bootstrap* phase, we propose the following experiment.

First, we run a *bootstrap* phase for $T = 10,000$ time steps to learn actions and their interests. Then, instead of the described *deployment* phase, we generate 20 random preferences as weight vectors $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$. We use a Dirichlet distribution, parameterized by $[1, 1, 1, 1]$, which returns vectors of norm 1. The automatic policies based on these weights take the action (p, a) that maximizes the scalarized interests: $(p, a) = \text{argmax}_{p,a} \sum_{i=1}^{m=4} \alpha_i \mathtt{Q}_{p,i}(\mathtt{States}_p(\vec{o}), a)$. We emphasize that such preferences are only used for this experiment: they are not necessary to our algorithm.

We create an alternative *deployment* phase, with one agent for each of the 20 automatic policies; they collect rewards for $T = 10,000$ steps. Policy scores are the average of rewards on each ethical objective, in $[0, 1]^m$. These scores are plotted in Fig. 2: each graph shows the policies' scores on 2 ethical objectives. They show that there are tradeoffs between objectives; e.g., the "orange" preference yields one of the highest scores on "well-being" and "affordability" (~ 0.8), but the lowest on "equity" (~ 0.4). All policies manage to attain a near-perfect score on at least one objective; yet, exploration could have been better. In particular, the "environmental" objective seems difficult to learn: all our policies' scores revolve around 0.2 on this one.

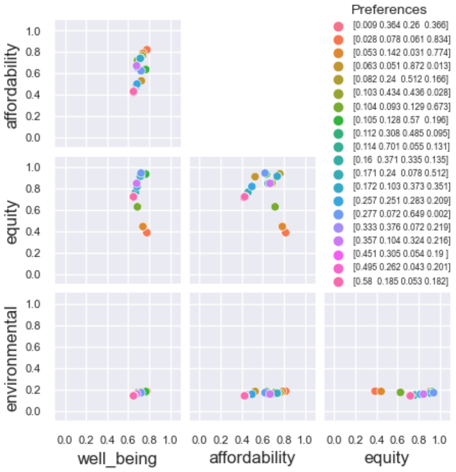

Fig. 2. Scores obtained by various policies (parametrized by the preferences in the legend) on all ethical objectives. Objectives are plotted 2-by-2: each point represents the score of a policy on 2 given objectives (X and Y axes).

We also computed the Hypervolume and Sparsity metric, with the same method as [21]; these measures highly depend on the used preferences, and should be taken with caution. The hypervolume was 0.103 (using $[0, 0, 0, 0]$ as the reference point), which is low, indicating that exploration could be improved. The sparsity was 0.023, which is better, meaning that there are no "gaps" in the Pareto Front. This highlights the *boostrap* phase capacity to learn various interesting actions.

B. Dilemmas are manageable and quickly learned

An important aspect of our work is the ease of use for humans. We want that: 1) each dilemma proposes a small set of actions from which to choose, to limit the cognitive load required to select an action; 2) the number of "remaining" dilemmas, i.e., dilemmas that remain to handle, diminishes with time. We hypothesize that, at the simulation start, as few contexts are known, most dilemmas will not correspond to any known context. As contexts are added, more dilemmas will be recognized, and this number should drop.

To confirm this, we performed a simulation with one of the authors as the human user, defining contexts and choosing actions when dilemmas are identified. After $T = 10,000$ time steps, we analyzed the contexts that were defined, knowing all the dilemmas that appeared during the simulation, and measured, for each context, how many dilemmas it recognized after its creation. This indicates how many future dilemmas are still "remaining", i.e., not recognized by any context, at any point of the simulation.

Fig. 3 shows that most of the contexts definitions happen at the beginning of the simulation. It is in line with our hypothesis, and the number of remaining dilemmas drastically

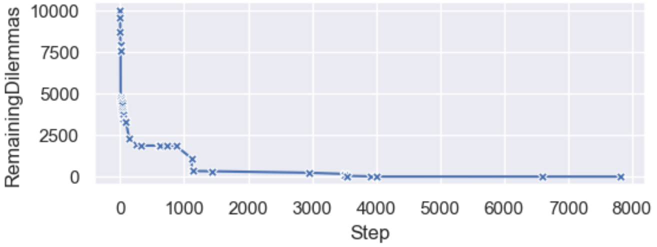

Fig. 3. Number of remaining dilemmas, not recognized by any context, during a simulation. Each cross indicates the creation of a new context.

drops during the 50 first time steps. The user was asked to define a new context 42 times, which is appreciable, compared to the 10,000 total time steps. Studies with several non-expert users are necessary to demonstrate the usability; yet, this suggests its feasibility, and reduced cognitive load.

In addition, the number of proposed actions in each dilemma was greatly reduced by using our filtering method. The total number of proposed actions was 45 each step (9 actions × 5 profiles); the Pareto Front reduces it to 5 to 28 actions (mean=16); removing similar actions in the PF finally reduces it to 2 to 7 actions (mean=4), which is far easier for users.

V. Discussion

A. Advantages of QSOM-MORL

In this paper, we have described a new MORL approach. Instead of asking the user for some numerical weights, we bring dilemmas to light and ask for intelligible and contextual preferences. This is in line with an *ethical companion* idea, enabling co-construction between the AI system and humans. The system learns to settle dilemmas from humans preferences, but humans may also learn from the system. They may discover dilemmas that they ignored, e.g., because of a lack of information, evaluate their preferences' impact, and potentially reconsider them. The "blocks" architecture allows improving specific parts, without compromising the entire concept.

B. Remaining perspectives and lines of research

The definitions we have proposed – ethical thresholds, acceptable actions, dilemmas – could be reworked, especially to take the user view into account, and improve human usability.

The "learning interesting actions" block could be replaced by more complex mechanisms, such as curiosity or intrinsic motivation [22], to explore the action space autonomously, rather than guiding through designer-specified weight vectors.

The "aggregation dilemmas" block could profit from a semi-automated approach. Dilemmas are discovered incrementally, and we have no single ground-truth as to which dilemmas belong to which group; thus, *unsupervised online clustering* approaches could help, by suggesting classifications. Users could accept or reject such suggestions, merge similar contexts together, or separating into a new group.

Because users preferences may evolve, or not stay coherent, the "learning user preferences" block should be updatable. Our algorithm supports such changing preferences, but it requires more reflection on the user interface part. It could also, in the long term, learn users' profiles automatically from various data, including moral questionnaires and everyday behaviors. This is close to the *ethics bot* idea [23], but requires systems capable of "grasping" the underlying moral values in the various data. It also raises serious issues: do we want a system that observes our every move to induce our moral values?

References

[1] A. Luccioni and Y. Bengio, "On the morality of artificial intelligence," *arXiv preprint arXiv:1912.11945*, 2019. 1

[2] J. H. Moor, "The nature, importance, and difficulty of machine ethics," *IEEE intelligent systems*, vol. 21, no. 4, pp. 18–21, 2006. 1

[3] M. Anderson and S. L. Anderson, *Machine ethics*. Cambridge University Press, 2011. 1

[4] A. W. Musschenga, "Empirical ethics, context-sensitivity, and contextualism," *J. Med. Philos.*, vol. 30, no. 5, pp. 467–490, 2005. 1

[5] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, *Handbook of computational social choice*, 2016. 1

[6] P. K. Murukannaiah, N. Ajmeri, C. M. Jonker, and M. P. Singh, "New foundations of ethical multiagent systems," in *AAMAS*, 2020. 1

[7] N. Cointe, G. Bonnet, and O. Boissier, "Ethical judgment of agents' behaviors in multi-agent systems." in *AAMAS*, 2016, pp. 1106–1114. 2

[8] P. Vamplew, R. Dazeley, C. Foale, S. Firmin, and J. Mummery, "Human-aligned artificial intelligence is a multiobjective problem," *Ethics and Information Technology*, vol. 20, pp. 27–40, 2018. 2

[9] Y.-H. Wu and S.-D. Lin, "A low-cost ethics shaping approach for designing reinforcement learning agents," in *Proc. of AAAI*, vol. 32, no. 1, 2018. 2

[10] M. Rodriguez-Soto, M. Serramia, M. Lopez-Sanchez, and J. A. Rodriguez-Aguilar, "Instilling moral value alignment by means of multi-objective reinforcement learning," *Ethics and Information Technology*, vol. 24, no. 1, p. 9, 2022. 2

[11] B. Alcaraz, O. Boissier, R. Chaput, and C. Leturc, "Ajar: An argumentation-based judging agents framework for ethical reinforcement learning," in *Proceedings of AAMAS*, 2023, pp. 2427–2429. 2, 5

[12] R. Chaput, O. Boissier, and M. Guillermin, "Adaptive reinforcement learning of multi-agent ethically-aligned behaviours: the qsom and qdsom algorithms," *arXiv preprint arXiv:2307.00552*, 2023. 2, 3, 5

[13] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992. 2

[14] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990. 2

[15] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker, "Empirical evaluation methods for multiobjective reinforcement learning algorithms," 2011. 2

[16] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Auton. Agent Multi Agent Syst.*, vol. 36, no. 1, 2022. 2, 5

[17] Y. Mukai, Y. Kuroe, and H. Iima, "Multi-objective reinforcement learning method for acquiring all pareto optimal policies simultaneously," in *Int. Conf. on Systems, Man, and Cybernetics*, 2012, pp. 1917–1923. 2

[18] T. Yamaguchi, S. Nagahama, Y. Ichikawa, and K. Takadama, "Model-based multi-objective reinforcement learning with unknown weights," in *Interacción*, 2019. 2

[19] H. Soh and Y. Demiris, "Evolving policies for multi-reward partially observable markov decision processes (mr-pomdps)," in *Proc. of the conf. on Genetic and evolutionary computation*, 2011, pp. 713–720. 2

[20] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966. 3

[21] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik, "Prediction-guided multi-objective reinforcement learning for continuous robot control," in *Proceedings of the ICML*, 2020. 5

[22] A. Aubret, L. Matignon, and S. Hassas, "An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey," *Entropy*, vol. 25(2):327, 2023. 6

[23] A. Etzioni and O. Etzioni, "Incorporating ethics into artificial intelligence," *The Journal of Ethics*, vol. 21, no. 4, pp. 403–418, 2017. 6