Bridging the Gap between Value Alignment and Multi-Objective Reinforcement Learning

Manel Rodriguez-Soto[1][0000−0003−1339−2018], Nardine Osman[1][0000−0002−2766−3475], and Jordi Sabater-Mir[1][https://orcid.org/0000−0001−6982−3572]

Artificial Intelligence Research Institute (IIIA-CSC), Barcelona, Spain
{manel.rodriguez,nardine,jsabater}@iiia.csic.es

Abstract. Designing protocols that align with organisational values, and training members of an organisation to respect those values is key for ensuring value-aligned behaviour. However, both problems are difficult, especially if there are multiple conflicting values that need to be respected. This work proposes to apply multi-objective reinforcement learning (MORL) to tackle both problems. First, we prove how current formal models of value-alignment are compatible with MORL models. Then, we present a novel process for computing and evaluating value-aligned protocols. Finally, we illustrate our protocol design process with an example scenario involving firefighters.

Keywords: Value Alignment · Multi-Objective Reinforcement Learning · Pareto front.

1 Introduction

The process of aligning a new member with the values of an existing group or organisation is of paramount importance. Many organisations already have written codes of ethics that enumerate their values, but it is often difficult to apply them in practice. Consider, for instance, a new firefighter in a fire department. The new member will need to make an effort and learn how their actions align with the department's values at the same time that they are working in a stressful environment. This situation will be prone to cause misaligned behaviour at times.

A solution that organisations have found is to develop protocols that establish how to behave in case of doubt. These protocols consist of a written set of norms that describe the approved, forbidden, and recommended actions or sequence of actions to perform under challenging decisions [6]. They are more specific and detail-oriented than codes of ethics because they specify who does 'what', 'when' and 'how' on each decision.

Both documents, codes of ethics and protocols should always be aligned with the same group values. For example, a fire department expects a firefighter protocol to align with values such as proximity, professionalism, or teamwork [1–3]. Once a protocol is established, new members must learn and apply it to

align with the organisational values. Typical approaches end here by proposing that the new members memorise all protocols.

However, developing protocols takes much work for organisations. Moreover, it is singularly difficult if the organisation's code of ethics contains multiple, possibly conflicting, values. As argued by Sierra *et al.* in [10], computing protocols aligned with multiple values is still an open problem. Protocols have a second associated open problem. Assume that an organisation reaches an agreement on its protocols. Then, its new members still have to learn them. However, it may be unrealistic to expect that these members will memorise all protocols by only reading them. Against this background, this paper presents a novel model for computing and teaching value-aligned protocols providing the following three contributions:

1. First, we prove how the value-alignment problem can be reformulated as a multi-objective reinforcement learning (MORL) problem. MORL literature deals with sequential decision-making problems in which agents need to manage multiple objectives [7, 8, 4].
2. Second, our theoretical results lead us to a novel process for computing the most value-aligned protocols for any simulated scenario with MORL algorithms.
3. Third, we present a reinforcement learning model to evaluate a group member's behaviour in any simulated scenario in terms of alignment with respect to all organisational values.

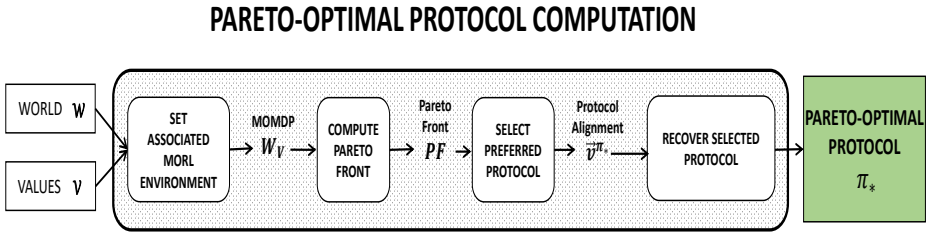

Fig. 1. Pareto-optimal protocol computation process design. Rectangles stand for objects whereas rounded rectangles correspond to processes.

The remainder of this paper is structured as follows. First, Section 2 offers the necessary background to understand our formal definitions of value-alignment and protocols. Then, Section 3 presents our novel process for computing and evaluating value-aligned protocols with respect to multiple values. Subsequently, Section 4 presents an example application of our model in a toy problem in a firefighters' context. Finally, Section 5 summarises the main contributions of this paper and sets paths for future work.

2 Background

2.1 Value-alignment

An agent (either software agent, robot agent, or human agent) is said to behave value-aligned if their actions promote a given human value.

Values can be abstract concepts such as *proximity* or *professionalism*. However, to compute whether a specific behaviour is value-aligned, we must first formalise value alignment. In this work, we follow the definitions of [10] and consider values as preferences that allow us to compare different states of the world.

The world is defined as the environment in which an agent can behave and whose actions have consequences. Formally [10]:

Definition 1 (World). *The world is defined as a labelled transition system* $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$*, where* $\mathcal{S}$ *is a set of states,* $\mathcal{A}$ *is a set of actions, and* $\mathcal{T}$ *is a set of labelled transitions* ($\mathcal{T} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$)

A value V, in a given world $\mathcal{W}$, then specifies which states of the world are preferred to which other states and to what degree. These preferences are numerically established using an alignment function *align*. The *align* function may compare two states s and s' resulting from a transition (s, a, s') [9], or may also take into account the action performed in the transition $a \in (s, a, s')$ [6]. We present here its most general form, evaluating the whole transition:

Definition 2 (Action alignment). *Let* $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ *be a world. The action alignment function align of* $\mathcal{W}$ *is defined as a function* $align : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{V} \rightarrow [-1, 1]$ *where* $\mathcal{V}$ *is a set of values. We want the range of alignment to be* $[-1, 1]$ *so that positive alignment would represent the action promoting the value, negative alignment would represent the action demoting the value, and an alignment of zero would represent the action not affecting the value.*

Moreover, an agent can repeatedly act upon the world, creating a sequence of transitions. This sequence is called a *path* [10]:

Definition 3 (Path). *A path* p *in a world* $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ *is a sequence of transitions* $p = \{T_i\}_i$*, with each* $T_i = (s_i, a_i, s'_i) \in \mathcal{T}$ *such that, for every* i*, it holds that* $s'_i = s_{i+1}$*. In other words, every transition's final state equals the following transition's initial state.*

Given a path p, we refer to the initial state S_i of each transition T_i of p as p_{s_i}, to its final state s'_i as $p_{s'_i}$, and to its action as p_{a_i}. For each path p of finite length, we can compute its alignment as the average alignment over all its transitions. Formally [10]:

Definition 4 (Finite path alignment). *Let* $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ *be a world. The path alignment function alignp of* $\mathcal{W}$ *is defined as a function* $alignp : \mathcal{P}_f \times \mathcal{V} \rightarrow$

$[-1, 1]$ *where* $\mathcal{P}_f$ *is the set of all possible paths of finite length in* $\mathcal{W}$*, and* $\mathcal{V}$ *is a set of values, such that:*

$$alignp(p, V) \doteq \frac{\sum_{i=0}^{length(p)} align(p_{s_i}, p_{a_i}, p_{s'_i}, V)}{length(p)}. \tag{1}$$

Given a world $\mathcal{W}$ with a set of actions $\mathcal{A}$, norms are a means to regulate the permitted behaviours of the agent interacting upon it. Each norm regulates an action a under a given state s of the world $\mathcal{W}$, indicating whether such action is permitted, obligatory, or forbidden. Formally:

Definition 5 (Norm). *Let* $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ *be a world. A norm* $N = D(a, s)$ *in* $\mathcal{W}$ *is defined as a tuple of one action* $a \in \mathcal{A}$*, and one state* $s \in \mathcal{S}$ *affected by one deontic operator* $D \in \{F, P, O\}$ *(where* F *describes what is forbidden,* P *what is permitted, and* O *what is obligatory).*

As mentioned, a given world $\mathcal{W}$ can be regulated by a set $\mathcal{N}$ of multiple norms. This set of norms $\mathcal{N}$ also establishes a *normative world* $\mathcal{W}_\mathcal{N}$ with a smaller subset of transitions $\mathcal{T}_\mathcal{N} \subseteq \mathcal{T}$. Given a set of norms $\mathcal{N}$, Sierra et al. in [9] defined the *norm alignment* $alignN(N, V)$ as the average path alignment over all possible paths in a normative world $\mathcal{W}_\mathcal{N}$ regulated by $\mathcal{N}$. Here, we consider an alternative definition for norm alignment, formalised in Definition 12.

Finally, to compute value alignment concerning a set of multiple values, we require an *aggregation function* $F_\mathcal{V} : [-1, 1]^n \to [-1, 1]$ that can be composed with the previous alignment functions to return a single scalar metric [9, 5]. This function $F_\mathcal{V}$ was left unspecified as future work. In this work, we offer more structure to the value aggregation function in the next Section 3.

2.2 Reinforcement learning

Reinforcement learning (RL) is the area of machine learning which formalises and aims to solve sequential decision-making problems [11]. In RL, an agent repeatedly interacts with an environment, called a *Markov decision process* (MDP), acting upon it, observing how the MDP transitions to a different state and receiving a reward signal. This reward signal is aligned with the agent's objective. In multi-objective reinforcement learning (MORL), the agent has multiple objectives, and thus, for each action, it receives multiple reward signals [7]. Formally:

Definition 6 (Multi-objective Markov decision process). *A (finite)* n*-objective Markov Decision Process (MOMDP) is defined as a tuple* $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{R}, T \rangle$ *where* $\mathcal{S}$ *is a (finite) set of states,* $\mathcal{A}(s)$ *is the set of actions available at state* s*, and* $\boldsymbol{R}$ *and* T *are functions defined as:*

- $\boldsymbol{R} = (R_1, \ldots, R_n)$*, the* vectorial *reward function with each* $R_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ *being the associated scalar reward function to objective* $i \in \{1, \ldots, n\}$*. For each objective* i*, the reward* $R_i(s, a, s')$ *indicates the goodness of applying action* $a \in \mathcal{A}$ *upon state* $s \in \mathcal{S}$ *if it transitions to state* s'*.*

– $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, *the transition function, returns the probability* $T(s, a, s') = \mathbb{P}(s' \mid s, a)$ *of* s' *being the next state if action* a *is applied upon sate* s.

In general, the probability of transitioning to a given state $s' \in \mathcal{S}$ and receiving a reward $r \in \mathbb{R}$ depends on the whole history of the agent (i.e., all states previously visited, all actions previously applied, and all rewards previously received). However, since in every MOMDP the *Markov property* is satisfied, we only need to care about the immediately previous state and action. Hence, the reward and transition functions are well-defined for every MOMDP.

In reinforcement learning, the behaviour of an agent upon its environment (the MOMDP), is formalised as a *policy*. Formally, for every state-action pair $\langle s, a \rangle$ of the MOMDP, a policy indicates the probability of performing action a upon state s:

Definition 7 (Policy). *Given an MOMDP* $\mathcal{M}$*, a policy* $\rho : \mathcal{A} \times \mathcal{S} \to [0, 1]$ *is a conditional probability defined as* $\rho(a, s) \doteq \mathbb{P}(a \mid s)$.

For every MOMDP $\mathcal{M}$, the policy ρ of the agent characterises the amount of rewards $\boldsymbol{R}$ that it will obtain by following it. For each objective i of the MOMDP $\mathcal{M}$, each policy's expected accumulation of reward is computed by its associated *value function* v_i. Formally:

Definition 8 (Vector value function). *Given a multi-objective MDP* $\mathcal{M}$*, and a policy* ρ*, its* vector value function $\boldsymbol{v}^\rho$ *returns the expected accumulation of vector rewards that* ρ *will obtain given an initial state* $s \in \mathcal{S}$*:*

$$\boldsymbol{v}^\rho(s) \doteq \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{R} \mid s_0 = s, \rho], \tag{2}$$

discounted by a discount factor $\gamma \in (0, 1)$*, indicating the importance of future rewards.*

The associated vector value function of any given policy can be computed with algorithms such as *policy evaluation* or *Monte Carlo prediction* [11].

In single-objective RL, the agent only has a single reward signal and thus only needs to maximise its associated value function. The optimal policy ρ_* is defined as the policy whose associated value function maximises the accumulation of rewards [11]:

Definition 9 (Optimal policy). *Given a single-objective MDP* $\mathcal{M}$*, its optimal policy* ρ_* *is the policy with maximum accumulation of discounted rewards:*

$$v^{\rho_*}(s) \doteq \max_\rho v^\rho(s), \tag{3}$$

for every state s *of* $\mathcal{M}$.

All algorithms aim to compute an optimal policy in the single-objective reinforcement learning literature. A classical algorithm is *value iteration* (VI) [11], which is guaranteed to always obtain an optimal policy in a finite amount of iterations.

In multi-objective reinforcement learning, there are multiple objectives to satisfy simultaneously. A policy that is optimal for all objectives rarely exists. Alternatively, the MORL literature has focused on learning all policies that are at least *Pareto-optimal* [7]. A policy ρ is Pareto-optimal if there is no alternative policy ρ' such that it improves ρ on all objectives. Notice that, in particular, all policies that are optimal for at least one objective are also Pareto-optimal. To illustrate the concept of Pareto-optimality, we provide a simple example:

Example 1. Consider the vectors $v_1 = (5, 0)$, $v_2 = (7, 2)$, $v_3 = (6, 3)$ in $\mathbb{R}^2$. The vector v_1 is not Pareto-optimal because $5 < 7$ and $0 < 2$. Meanwhile, both v_2 and v_3 are Pareto-optimal because $7 > 6$ and $2 < 3$.

In MORL, the set of Pareto-optimal policies π_* and their associated value vectors $\boldsymbol{V}^{\pi_*}$ receives the name of *Pareto front* [7].

The multi-objective equivalent of the Value Iteration algorithm is *Pareto Multi-Objective Value Iteration* (PMOVI) [7]. While PMOVI cannot obtain policies, it is guaranteed to obtain the associated value functions $\boldsymbol{v}$ for all Pareto-optimal policies in a finite amount of iterations. Afterwards, the Pareto-optimal policies can be recovered by the *so-called* policy-tracking-process from [12, 7].

3 Value-aligned protocols

This section presents our multi-objective reinforcement learning model for evaluating and computing protocols while considering multiple values. This Section is structured as follows. First, Section 3.1 provides a formal definition of a protocol aligned with multiple values, which extends the previously presented *path alignment* definition. After that, Section 3.2 proves that our protocol alignment definition can be expressed in terms of MORL concepts. Finally, exploiting this theoretical result, Section 3.3 details our process for computing value-aligned protocols with MORL algorithms.

3.1 Formalising value-alignment for multiple values

As previously explained, protocols are sets of norms that all members of an organisation must follow. We first require a formalisation of protocol alignment based on the action alignment definition to compute if a given protocol is aligned or misaligned.

While protocols are typically defined simply as sets of norms [6], we provide a more general definition in this work. We consider *protocols* of worlds analogously to policies of MOMDPs. A protocol tells us how much recommended is a given action for every possible state of the world $\mathcal{W}$. Hence, for every state-action pair $\langle S, A \rangle$, any protocol π returns a number between 0 and 1, indicating the degree

of recommendation of the action. A 0 indicates that the action A is forbidden at state S, while the other extreme 1 indicates that the action A is obligatory at state S. Anything in between indicates that the action A is permitted at state S (and how much it is recommended) but not obligatory (i.e., at least one alternative action A' permitted in the currently considered state S). Formally:

Definition 10 (Protocol). *Let $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ be a world. A protocol $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$ comprises the recommended actions that an agent should do for every possible state $S \in \mathcal{S}$.*

Given our definition of protocol, we will say that a protocol π includes a given norm N if and only if:

- $N = F(a, s)$ and $\pi(a, s) = 0$,
- $N = O(a, s)$ and $\pi(a, s) = 1$, and
- $N = P(a, s)$ and $\pi(a, s) \in (0, 1)$.

Notice how our definition of protocol is more expressive than a set of norms. Given a state-action pair $\langle s, a \rangle$, the degree of recommendation $0 \leq \pi(a, s) \leq 1$ can also be understood as the probability of an agent following it to perform action A at state S. Thus, the value-alignment of a protocol π can be directly formalised as the expected average action alignment that an agent following π would receive. By abuse of notation, in this paper, we also formalise the behaviour of any agent as a protocol π.

We first require an auxiliary term: *path alignment*, to measure protocol alignment. Recall that path alignment for paths of finite length has already been formalised according to Definition 4. However, one may think of many situations in which an agent is expected to continuously apply a protocol without a clear end. In other words, we also need to consider paths of possibly infinite length. Following the reinforcement learning literature [11], we consider that, when following a path of possibly infinite transitions, we need to evaluate future actions less importantly than current ones. Analogously to RL, the degree of importance of future transitions is set by means of a discount factor $\gamma \in (0, 1)$, such that the greater γ, the more we care about future events.

Given a discount factor γ, we define the path alignment of a given path of arbitrary length as the *discounted* accumulation of alignment that following such path would entail. Formally:

Definition 11 (Path alignment). *Let $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ be a world. Its path alignment function $alignp_\gamma$ is defined as a function $align_\gamma : \mathcal{P} \times \mathcal{V} \to [-1, 1]$ where $\mathcal{P}$ is the set of all possible paths in $\mathcal{W}$, $\gamma \in (0, 1)$ is the discount factor, and $\mathcal{V}$ is a set of values, such that:*

$$alignp_\gamma(p, V) \doteq (1 - \gamma) \sum_{i=0}^{\infty} \gamma^i \cdot align(p_{s_i}, p_{a_i}, p_{s'_i}, V). \tag{4}$$

Given a discount factor γ, Definition 11 multiplies the obtained alignment by $(1 - \gamma)$. This normalisation factor is applied to guarantee that $alignp_\gamma$ always

returns a number between 0 and 1.[1] By abuse of notation, we will refer to the path alignment function of paths of finite length *alignp* also as $alignp_1$ (i.e., corresponding to a discount factor $\gamma = 1$).

Definition 11 of path alignment allows us to formalise a protocol alignment function as the average path alignment that an agent would obtain by following a protocol π for all possible paths that start at a given initial state S. Formally:

Definition 12 (Protocol alignment). *Let $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ be a world. Let $\gamma \in (0, 1]$ be a real number. The protocol alignment function $alignP_\gamma$ of $\mathcal{W}$ is defined as a function $alignP_\gamma : \Pi \times \mathcal{S} \times \mathcal{V} \to [-1, 1]$ where $\mathcal{V}$ is a set of values such that:*

$$alignP_\gamma(\pi, S, V) \doteq \sum_{p \in \mathcal{P}_S} \mathbb{P}(p \mid \pi) \cdot alignp_\gamma(p, V), \tag{5}$$

where Π is the set of possible protocols, $\mathcal{P}_S$ is the set of paths of $\mathcal{W}$ such that the initial state s_0 of their initial transition $T_0 = (s_0, a_0, s_0')$ is $s_0 = s$, and $\mathbb{P}(p \mid \pi)$ is the probability of each path p of $\mathcal{P}_S$ occurring subject to protocol π.

Definition 12 provides us with a tool to evaluate, and optimise protocols with respect to values. In particular, given the action alignment function $align(\cdot, \cdot, V)$ of a value V, for every initial state s of the world $\mathcal{W}$, we can define its optimal protocol π_V as the protocol that obtains the maximum amount of protocol alignment. Formally:

Definition 13 (Optimal protocol). *Let $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ be a world, let Π be the set of possible protocols of $\mathcal{W}$, and let V be a value. We define the optimal protocol π_V of $\mathcal{W}$ according to the value V at state $s \in \mathcal{S}$ as:*

$$\pi_V \doteq \max_{\pi \in \Pi} alignP(\pi, s, V). \tag{6}$$

Definition 13 provides us with a way of defining the most value-aligned protocol with respect to a single value. However, as we have previously argued, many organisations consider multiple values. Regarding our formalisation, the ideal protocol would need to be optimal with respect to all values.

Since not all actions are always equally aligned to all values, we require a trade-off between values. As explained in Section 2, Sierra *et al.* in [10] argued that in such cases, we require a value aggregation function capable of computing the alignment of each action weighing the importance of all considered values.

However, defining the optimal value aggregation function f_* that perfectly represents the preferences between the different values is a difficult task. We can assume that at least this function will be strictly monotonically increasing. That is, when comparing two protocols, we will always prefer the one that Pareto-dominates the other in terms of protocol alignment. For this reason, following the MORL literature, we aim to find the set of protocols that cannot be Pareto-dominated (i.e., that are Pareto-optimal). Formally:

[1] Notice that $\sum_{i=0}^{\infty} \gamma^i = \frac{1}{1-\gamma}$ for any $\gamma \in (0, 1)$.

Definition 14 (Pareto-optimal protocol). *Let $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ be a world, let Π be the set of possible protocols of $\mathcal{W}$, and let $\mathcal{V}$ be a finite set of values. A protocol $\pi_* \in \Pi$ is Pareto-optimal at state $s \in \mathcal{S}$ if and only if, for every other protocol $\pi \in \Pi$, if $alignP(\pi, s, V_i) > alignP(\pi_*, s, V_i)$ for some value $V_i \in \mathcal{V}$, then $alignP(\pi, s, V_i) < alignP(\pi_*, s, V_i)$ for another value $V_j \in \mathcal{V}$.*

Our goal is to find, given the initial state s_0 of a world $\mathcal{W}$, its set of Pareto-optimal protocols Π_*. Then, amongst these Pareto-optimal protocols, the organisations' decision-makers can select which one they prefer. Before entering into details on how to compute Pareto-optimal protocols. Next Section 3.2 presents our process for computing Pareto-optimal protocols.

3.2 Value-aligned protocols computation

Recall that our goal is to obtain a set of candidate protocols for the decision-maker, so they can later decide the one they prefer. All these candidate protocols will be Pareto-optimal. Thus, we require a process for computing all Pareto-optimal protocols given a world $\mathcal{W}$.

Our present process transforms our value-alignment problem into a multi-objective reinforcement learning problem. We first prove that such conversion is possible and how. After that, we provide a process guaranteed to obtain Pareto-optimal protocols with MORL algorithms.

In the remainder of this Section we assume that the considered $\mathcal{W}$ satisfies the Markov property on its transitions. Formally:

Assumption 1 *The world $\mathcal{W} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$ is such that for every $t \in \mathbb{N}$:*

$$\mathbb{P}(S_{t+1} = S' \mid S_t, A_t, S_{t-1}, A_{t-1}, \ldots, s_0, a_0) = \tag{7}$$

$$\mathbb{P}(S_{t+1} = S', \mid S_t, A_t). \tag{8}$$

Assumption 1 is satisfied in a wide range of worlds. In particular, any deterministic world (i.e., a world where applying an action a to a state s always transitions to the same state s') fulfils Assumption 1.

For any world $\mathcal{W}$ satisfying Assumption 1, given a value V, we can readily create an associated Markov decision process $\mathcal{W}_V$. In $\mathcal{W}_V$ its sets of states and actions, and the transition function are determined by $\mathcal{W}$, and the reward function is the associated action alignment function $align(\cdot, \cdot, V)$ of V. If we consider multiple values, we can generalise this procedure to create an associated MOMDP. Formally:

Lemma 1. *Let $\mathcal{V} = (V_1, \ldots, V_n)$ be a set of n values. Let $\mathcal{W} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$ be a world satisfying Assumption 1. Then, the tuple $\mathcal{W}_\mathcal{V} = \langle \mathcal{S}, \mathcal{A}, T, align(\cdot, \cdot, V_1), \ldots, align(\cdot, \cdot, V_n)$ is a multi-objective Markov decision process, where T is a transition function determined by the transition set $\mathcal{T}$, and $align(\cdot, \cdot, V_i)$ is the action alignment function of value V_i.*

The natural conclusion from Lemma 1 is that we can also draw a parallelism between Pareto-optimal policies and Pareto-optimal protocols. In fact, next Theorem 1 proves that, as long as Assumption 1 holds, any Pareto-optimal protocol π_* of the world $\mathcal{W}$ is also a Pareto-optimal policy of the MOMDP $\mathcal{W}_\mathcal{V}$ and vice-versa. Formally:

Theorem 1. *Let $\mathcal{V}$ be a set of values and $\mathcal{W}$ a world that fulfils Assumptions 1. Let $\mathcal{W}_\mathcal{V}$ be the associated MOMDP of $\mathcal{W}$ and $\mathcal{V}$. Set the discount factor γ smaller than 1. Then, any protocol in $\mathcal{W}$ is a Pareto-optimal if and only if it is also a Pareto-optimal policy in $\mathcal{W}_\mathcal{V}$.*

Proof. It suffices to prove that, for each value V_i, the protocol alignment function $alignP_\gamma(\pi, s, V_i)$ with $\gamma < 1$ and the value function $v_i^\pi(s)$ are proportional in $\mathcal{W}$ if Assumption 1 holds:

$$alignP_\gamma(\pi, s, V_i) = \tag{9}$$

$$\sum_{p \in \mathcal{P}_S} \mathbb{P}(p \mid \pi) \cdot alignp_\gamma(p, V_i) = \tag{10}$$

$$\mathbb{E}[(1-\gamma) \cdot \sum_{i=0}^{\infty} \gamma^i \cdot align(p_{s_i}, p_{a_i}, p_{s'_i}, V_i) \mid s_0 = s, \pi] = \tag{11}$$

$$(1-\gamma) v_i^\pi(s). \tag{12}$$

Notice that the proof of Theorem 1 teaches us how to evaluate any protocol. We can directly apply any reinforcement learning algorithm designed for computing the associated vector value function of a policy π (such as Monte Carlo prediction or policy evaluation) to obtain the protocol alignment of any protocol $alignP$. Formally:

Corollary 1. *Let $\mathcal{V}$ be a set of values and $\mathcal{W}$ a world that fulfils Assumptions 1. Let $\mathcal{W}_\mathcal{V}$ be the associated MOMDP of $\mathcal{W}$ and $\mathcal{V}$. Set the discount factor γ smaller than 1. Then, for any protocol π, and every value $V_i \in \mathcal{V}$, for every state s:*

$$alignP_\gamma(\pi, s, V_i) = (1-\gamma) v_i^\pi(s). \tag{13}$$

3.3 Protocols Pareto-optimisation process

Our proposed process for computing Pareto-optimal protocols consists of the following four steps:

1. First, given a set of values $\mathcal{V}$ and a world $\mathcal{W}$, we compute the associated MOMDP $\mathcal{V}_\mathcal{W}$ in which MORL algorithms can be applied.
2. Then, we feed the MOMDP $\mathcal{V}_\mathcal{W}$ as input for any MORL algorithm to compute the Pareto front PF of the associated protocol alignment $\boldsymbol{v}^{\pi_*}$ of all Pareto-optimal protocols $\pi_* \in \Pi_*$. In our case, we apply PMOVI.

Algorithm 1 Pareto-optimal protocol computation

Input: World $\mathcal{W} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$, action alignment functions $align(\cdot, \cdot, V_1), \ldots, align(\cdot, \cdot, V_n)$.

1: Set $\mathcal{W}_\mathcal{V}$ as the associated MOMDP of $\mathcal{W}$ and $\mathcal{V}$ as defined in Lemma 1.
2: Apply PMOVI [7] to $\mathcal{W}_\mathcal{V}$ to obtain the Pareto Front PF of the MOMDP $\mathcal{W}_\mathcal{V}$.
3: Select one of the Pareto-optimal values $\boldsymbol{v}^{\pi_*}$ from PF.
4: Apply policy-tracking-process [12] to compute the policy π_* associated with the selected value $\boldsymbol{v}^{\pi_*}$.
5: **return** Pareto-optimal protocol π_*.

3. After that, the stakeholders in charge of deciding the protocol select their preferred protocol π_* based on their respective protocol alignment $\boldsymbol{v}^{\pi_*}$ from the Pareto Front PF.
4. Finally, we need to use an MORL algorithm to recover the protocol π_* from its protocol alignment $\boldsymbol{v}^{\pi_*}$. In our case, we apply the policy-tracking-process of [12].

Algorithm 1 implements our four-step process to obtain the desired Pareto-optimal protocol. It receives as an input both a world $\mathcal{W}$ and the action alignment functions *align* for all values $\mathcal{V}$ that need to be taken into consideration.

4 Experimental analysis

The purpose of this section is twofold. First, it illustrates how to evaluate a protocol according to our protocol alignment function with an example. Second, it also shows an example application of our process for computing value-aligned protocols, as detailed in Algorithm 1. Due to the lack of benchmark environments that consider several values in MORL, we propose a novel firefighters environment which includes the values of *proximity* and *professionalism*[2].

This section is structured as follows. First, Section 4.1 presents the formalisation of the world $\mathcal{W}$ that simulates a firefighters' scenario. After that, Section 4.2 formalises the relevant values in this scenario (proximity and professionalism) and details their respective alignment functions. We proceed in Section 4.3 by showing an example of evaluating a given protocol by applying our definition of protocol alignment in this scenario. Finally, 4.4 applies Algorithm 1 to this firefighters' scenario to obtain a Pareto-optimal protocol.

4.1 World specification

We model this firefighters' scenario as a world $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$ with a finite amount of states and actions.

[2] Implementation code: https://github.com/Lenmaz/VALE2024-Firefighters/

State Space Each state $s \in \mathcal{S}$ of the environment is defined as a tuple (f, o, e, v, m), where each state variable represents:

- f: **Fire intensity** (None, Low, Moderate, High, Severe). Indicates the severity of the fire at the current state.
- o: **Occupancy** (0 to 4 people). Indicates whether there are people to be rescued and how many.
- e: **Equipment readiness** (Ready, Not Ready). Indicates if the firefighters have all the equipment in the current state to advance safely.
- v: **Visibility** (Good, Poor). Represents the environmental condition affecting firefighting efforts.
- m: **Medical condition of the firefighter** (incapacitated, moderately injured, slightly injured, perfect health). Shows if the firefighter can continue or not and how endangered they are.

There is a total of $5 \cdot 5 \cdot 2 \cdot 2 \cdot 4 = 400$ states. Some of them are terminal states (i.e., they do not transition to any other state). There are two separate groups of terminal states:

- States in which the medical condition of the firefighter is `incapacitated`.
- States in which there is occupancy 0 and no fire intensity.

The initial state of all paths is S_0 =`(Moderate fire intensity, four occupants, Equipment Not Ready, Poor Visibility, Perfect health)`.

Action Space and transitions The action space $\mathcal{A}$ consists of 5 actions:

- **Evacuate occupants:** Reduces the current level of `occupancy` by 1. If the action is performed under both `poor visibility` and `not ready equipment`, and also a `moderate fire intensity` or worse, the firefighter's `medical condition` gets reduced by 1. If the action is performed under a `severe fire intensity`, the equipment becomes `not ready`.
- **Contain fire:** Reduces the level of `fire intensity` by 1.
- **Aggressive fire suppression:** Reduces the current level of `fire intensity` by 2. If the action is performed under either `poor visibility` or `not ready equipment`, and also a `moderate fire intensity` or worse, the firefighter's `medical condition` gets reduced by 1. Also, if the action is performed under a `severe fire intensity`, the equipment becomes `not ready`.
- **Coordinate with other agencies:** Sets the equipment as `ready`.
- **Assess and plan:** Sets the visibility as `good`.

4.2 Firefighters' values specification

We consider two firefighters' values for this scenario: professionalism and proximity:

- **Professionalism:** The degree of agreement with the organisation's rules and principles of action.

– **Proximity:** The degree of knowledge and understanding of the society and the territory, and how the incident impacts them.

Professionalism dictates adherence to organisational standards and principles, which might involve following strict protocols that prioritise firefighter safety over aggressive firefighting tactics. However, proximity emphasises understanding the societal and territorial impact of the incident, pushing for a more aggressive approach to save lives and property, knowing the significant impact a complete burnout would have on the community. The conflict arises when the need to act according to official safety protocols conflicts with the urgent community-focused response demanded by the situation.

Alignment Functions This scenario contemplates two firefighter values: Proximity V_{prox} and Professionalism V_{prof}. Each value has its corresponding action alignment function $align$. We proceed to formalise both alignment functions for each possible action.

– **Evacuate occupants:**
  • **The professionalism alignment** is computed as $max(0, 1 - 0.2f - 0.1v)$. This implies a reduction in professionalism with increasing fire intensity and reduced visibility, reflecting the escalating risk and difficulty in managing the situation professionally under deteriorating conditions. However, if no occupants are present $o = 0$, a significant penalty is applied $align = -1$, indicating a gross misjudgment in evacuating an empty area.
  • **The proximity alignment** is $align = 1$ if there is at least one occupant needing to be rescued. If no occupants are present $o = 0$, a significant penalty is also applied $align = -1$.
– **Contain fire:** For both values, it returns an negative alignment $align = -1$ when trying to contain fire if there is `none` left. Otherwise:
  • **Professionalism alignment:** 0.8, since it is highly aligned with the value of professionalism.
  • **Proximity alignment**: 0.2, since it is moderately aligned with the value proximity.
– **Aggressive fire suppression:** For both values, it returns an negative alignment $align = -1$ when trying to contain fire if there is `none` left. Otherwise:
  • **Professionalism alignment:** 0.6, since it is aligned with the value of professionalism, but not as much as if it normally contained fire. The alignment goes down to $align = 0.3$ if the action is performed with `Equipment Not Ready`.
  • **Proximity alignment**: 0.5, since it is aligned with the value proximity.
– **Coordinate with other agencies:** For both values, it returns an negative alignment $align = -1$ when trying to coordinate when the equipment is already `ready`. Otherwise:
  • **Professionalism alignment:** 0.5, since it is aligned with the value of professionalism.

  - **Proximity alignment**: -0.1, since it is not aligned with the value proximity, and even slightly misaligned because the firefighter is losing time to rescue the people or suppress the fire.
- **Assess and plan:** For both values, it returns an negative alignment $align = -1$ when trying to assess when the visibility is already `good`. Otherwise:
  - **Professionalism alignment:** 1, since it is totally aligned with the value of professionalism.
  - **Proximity alignment**: -0.5, since it is highly misaligned with the value proximity. The firefighter is losing time to rescue the people or extinguish the fire.

Finally, if any transition $\langle s, a, s' \rangle$ goes to the terminal state S' with the firefighter `incapacitated`, both alignment functions return $align(s, a, s', V) = -1$, since it will allow the firefighter to neither rescue the people nor extinguish the fire.

4.3 Protocol evaluation

Having formalised the world $\mathcal{W}$ and values $\mathcal{V}$ of our scenario, we can now evaluate protocols within it. Given any protocol π, we can apply Monte Carlo prediction to compute its degree of protocol alignment with respect to this world $\mathcal{W}$. Monte Carlo predictions work by averaging the protocol alignment obtained in multiple path executions.

For this section we consider the protocol π that: (1) it obligates to always **assess and plan** prior to doing anything else; then, (2) it obligates to always **contain fire** completely before rescuing anyone; and finally, (3) it Obligates to **evacuate occupants** if there is no fire left. This protocol would produce the following path execution p:

1. At initial state S_0, it applies action **assess and plan**, receiving an alignment of 1 for **professionalism** and -0.5 for **proximity.**
2. At the following states S_1, S_2 and S_3, the agent applies action **contain fire** to suppress the fire. It receives an alignment of 0.8 for **professionalism** and 0.2 for **proximity** the three times. Fire intensity gets reduced from 3 to 0.
3. At the following state S_4, S_5, S_6, and S_7, the agent applies action **evacuate occupants** again to safely evacuate all occupants. It receives an alignment of 0.9 for **professionalism** because there is no fire left. and 1.0 alignment for **proximity.**
4. The world finally reaches a terminal state because there are no occupants left to be rescued and the fire is extinguished.

Since all transitions are deterministic in our example world $\mathcal{W}$, we can directly compute the protocol alignment of π from the path p. We apply now the equation from Definition 12 to obtain the protocol alignment of π:

$$alignP_{0.7}(\pi, S_0, V_{prof}) = (1 - 0.7) \cdot 2.77 = 0.83, \tag{14}$$

$$alignP_{0.7}(\pi, S_0, V_{prox}) = (1 - 0.7) \cdot 0.41 = 0.12. \tag{15}$$

With this process, any firefighter can evaluate their behaviour and assess how value-aligned it is to their organisation's values.

4.4 Pareto-optimal protocols computation

Setting the associated MOMDP First, we considered the world $\mathcal{W} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$ defined in Section 4.1, and the two alignment functions from values $\mathcal{V} = \{V_{prox}, V_{prof}\}$ defined in Section 4.2. We combined these elements (world and alignment functions) to set the associated MOMDP

$$\mathcal{W}_{\mathcal{V}} = \langle \mathcal{S}, \mathcal{A}, T, align(\cdot, \cdot, V_{prox}), align(\cdot, \cdot, V_{prof}) \rangle$$

in which MORL algorithms can be applied.

Computing the set of Pareto-optimal policies The next step of our process consisted of applying the MORL algorithm PMOVI to our MOMDP $\mathcal{V}_{\mathcal{W}}$ to compute the associated alignment of all Pareto-optimal policies. PMOVI is guaranteed to converge in a finite amount of iterations. PMOVI requires one hyperparameter: the discount factor γ. This discount factor is exactly the same one we will set for our protocol alignment function $alignP_{\gamma}$. In our case, we selected $\gamma = 0.7$. Afterwards, PMOVI needed 4 iterations to converge. For the initial state S_0, it computed 50 different Pareto-optimal protocols. Figure 2 shows the 50 protocols.

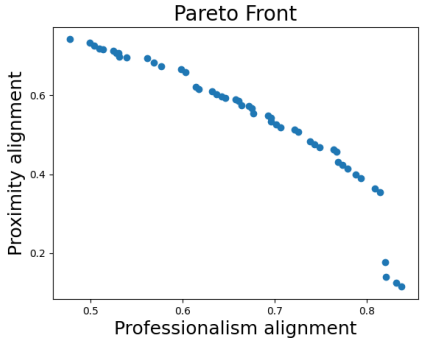

Fig. 2. Pareto front of the firefighters' scenario. Each point corresponds to the normalised value vector $\boldsymbol{v}$ of a Pareto-optimal protocol (i.e., its protocol alignment).

Selecting a policy In the next step, we would require the decision-maker to select from the list of 50 candidate Pareto-optimal protocols the one they prefer, strictly focusing on their value alignment.

For example, we assume that the decision-maker selected the protocol with the maximum amount of *proximity* alignment $\pi_* = \pi_1$, with an alignment of $(0.48, 0.74)$. (left-most protocol in Figure 2).

Recovering the policy from its value Finally, we applied the policy-tracking-process to recover π_* from its value $\boldsymbol{v}^{\pi_*}(S_0)$. This tracking process consists of exploiting the Bellman equation, that all policies satisfy:

$$\boldsymbol{v}^{\pi_*}(s) = \sum_{a \in \mathcal{A}} \pi_*(a, s)[\sum_{s' \in \mathcal{S}} T(s, a, s') \cdot (\boldsymbol{R}(s, a, s') + \gamma \cdot v^{\pi_*}(s'))]. \quad (16)$$

The Bellman equation allows us to recursively express the value of any policy (protocol). The policy-tracking-process works as follows:

First, for every state S, it computes all possible transitions t that have as a starting state S_0. This information can be directly accessed from the transition function if available or estimated via Monte Carlo prediction. Then, the policy-tracking-process aims to find the actions satisfying the Bellman equation in Equation 16 for every state.

However, the are two unknowns in Eq. 16: the probability $\pi_*(a, s)$ of selecting each action a by π_* at state s, and $\boldsymbol{v}^*(s')$, and an unknown value vector $\boldsymbol{v}^*(s')$ from the set of value vectors $\boldsymbol{V}^*(s')$ computed by PMOVI for state s'. To solve Equation 16, we search for the action a and value vector $\boldsymbol{v}^*(s')$ that satisfy the following equation:

$$\min_{a \in \mathcal{A}, \boldsymbol{v}^*(s') \in \boldsymbol{V}^*(s')} ||\boldsymbol{v}^{\pi_*}(s) - \sum_{s' \in \mathcal{S}} T(s, a, s') \cdot (\boldsymbol{R}(s, a, s') + \gamma \cdot \boldsymbol{v}^*(s'))|| = 0. \quad (17)$$

Solving Equation 17 provides us with the action that protocol π_* applies at state s. This way, we could recover the policy π_*. To illustrate, we show the policy-tracking-process for the initial state S_0:

- First, we computed all five transitions t starting at state S_0, one per action. We denote the final state of each of them as $S'_1, \ldots, S'_5$.
- For each of them, we extracted their respective value vectors $\boldsymbol{V}^*(S'_1), \ldots, \boldsymbol{V}^*(S'_5)$ computed by PMOVI.
- Afterwards, we computed the left side of Eq. 17 for each action a:
  1. For action **evacuate occupants**: 0.
  2. For action **contain fire**: 0.6.
  3. For action **aggressive fire suppression**: 0.21.
  4. For action **coordinate with other agencies**: 0.69.
  5. For action **assess and plan**: 1.06.
- The policy-tracking-process therefore sets $\pi_*(S_0) =$ **evacuate occupants** since it satisfied Equation 17. We obtained the whole protocol π_* by repeating this process for each state.

5 Conclusions

This paper has tackled the open problem of designing and evaluating protocols that align with multiple values. Our novel contributions are based on the framework of multi-objective Markov decision processes (MOMDP). First, we

have proven that value-aligned protocols can be expressed in terms of Pareto-optimal policies of MOMDPs. Based on our theoretical findings, we have provided a four-step process for computing value-aligned protocols that computes the Pareto Front of an MOMDP. Similarly, our theoretical findings pave the way for evaluating protocols, as we have proven that they can be evaluated with the same reinforcement learning tools to evaluate policies. In future work, we expect to analyse the effect of our process on the protocols of real organisations.

Acknowledgments. This work has been supported by the EU-funded VALAWAI (# 101070930) project and the Spanish-funded VAE (# TED2021-131295B-C31), EVA-SAI (# TED2021-131295B-C31 and # PID2024-158227NB-C31), and EMOROBCARE (# IASOMMA2024) projects.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. de Barcelona, B.: Bombers de Barcelona: Values. https://ajuntament.barcelona.cat/bombers/en/values (2016), [Online; accessed 11-June-2024]
2. de la Generalitat de Catalunya, B.: Bombers de la Generalitat de Catalunya: Carta de serveis. https://interior.gencat.cat/web/.content/home/010_el_departament/carta_de_serveis/docs_bombers/document_integre_cs.pdf (20009), [Online; accessed 11-June-2024]
3. Fire Department, C.o.N.Y.: Get to know the FDNY. https://www.joinfdny.com/about/ (2024), [Online; accessed 11-June-2024]
4. Hayes, C.F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L.M., Dazeley, R., Heintz, F., Howley, E., Irissappane, A.A., Mannion, P., Nowé, A., Ramos, G., Restelli, M., Vamplew, P., Roijers, D.M.: A practical guide to multi-objective reinforcement learning and planning. Autonomous Agents and Multi-Agent Systems **36** (2022)
5. Montes, N., Sierra, C.: Value-guided synthesis of parametric normative systems. In: Dignum, F., Lomuscio, A., Endriss, U., Nowé, A. (eds.) AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021. pp. 907–915. ACM (2021). https://doi.org/10.5555/3463952.3464060, https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p907.pdf
6. Rodriguez-Soto, M., Osman, N., Sierra, C., Veja, P.S., Garcia, R.C., Danes, C.F., Retortillo, M.G., Maso, S.M.: Towards value awareness in the medical field. In: Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 3: AWAI. pp. 1391–1398. INSTICC, SciTePress (2024). https://doi.org/10.5220/0012588600003636
7. Roijers, D., Whiteson, S.: Multi-Objective Decision Making. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, California, USA (2017), http://www.morganclaypool.com/doi/abs/10.2200/S00765ED1V01Y201704AIM034, doi:10.2200/S00765ED1V01Y201704AIM034
8. Rădulescu, R., Mannion, P., Roijers, D.M., Nowé, A.: Multi-objective multi-agent decision making: a utility-based analysis and survey. Autonomous Agents and Multi-Agent Systems **34**, 1–52 (2019)

9. Sierra, C., Osman, N., Noriega, P., Sabater-Mir, J., Perello-Moragues, A.: Value alignment: A formal approach. Responsible Artificial Intelligence Agents Workshop (RAIA) in AAMAS 2019 (2019)
10. Sierra, C., Osman, N., Noriega, P., Sabater-Mir, J., Perelló, A.: Value alignment: a formal approach (10 2021)
11. Sutton, R.S., Barto, A.G.: Reinforcement learning - an introduction. Adaptive computation and machine learning, MIT Press (1998), http://www.worldcat.org/oclc/37293240
12. Van Moffaert, K., Drugan, M., Nowe, A.: Multi-objective reinforcement learning using sets of pareto dominating policies. vol. 15, p. 1 (01 2013)