Making Human-Like Moral Decisions

Andrea Loreggia
University of Brescia
Brescia, Italy
andrea.loreggia@unibs.it

Nicholas Mattei
Tulane University
New Orleans, United States
nsmattei@tulane.edu

Taher Rahgooy
University of West Florida
United States
taher.rahgooy@gmail.com

Francesca Rossi
IBM Research
Yorktown Heights - NY, United States
francesca.Rossi2@ibm.com

Biplav Srivastava
University of South Carolina
Columbia, United States
biplav.s@sc.edu

Kristen Brent Venable
IHMC and UWF
United States
bvenable@uwf.edu

ABSTRACT

Many real-life scenarios require humans to make difficult trade-offs: do we always follow all the traffic rules or do we violate the speed limit in an emergency? In general, how should we account for and balance the ethical values, safety recommendations, and societal norms, when we are trying to achieve a certain objective? To enable effective AI-human collaboration, we must equip AI agents with a model of how humans make such trade-offs in environments where there is not only a goal to be reached, but there are also ethical constraints to be considered and to possibly align with. These ethical constraints could be both deontological rules on actions that should not be performed, or also consequentialist policies that recommend avoiding reaching certain states of the world. Our purpose is to build AI agents that can mimic human behavior in these ethically constrained decision environments, with a long term research goal to use AI to help humans in making better moral judgments and actions. To this end, we propose a computational approach where competing objectives and ethical constraints are orchestrated through a method that leverages a cognitive model of human decision making, called *multi-alternative decision field theory (MDFT)*. Using MDFT, we build an orchestrator, called MDFT-Orchestrator (MDFT-O), that is both general and flexible. We also show experimentally that MDFT-O both generates better decisions than using a heuristic that takes a weighted average of competing policies (WA-O), but also performs better in terms of mimicking human decisions as collected through Amazon Mechanical Turk (AMT). Our methodology is therefore able to faithfully model human decision in ethically constrained decision environments.

CCS CONCEPTS

• **Theory of computation** → **Reinforcement learning**; **Markov decision processes**; **Sequential decision making**; • **Computing methodologies** → *Theory of mind*; • **Human-centered computing** → HCI design and evaluation methods.

AIES '22, August 1–3, 2022, Oxford, United Kingdom.

ACM ISBN 978-1-4503-9247-1/22/08…$15.00
https://doi.org/10.1145/3514094.3534174

KEYWORDS

Ethical constraints, Human decision-making process, Markov Decision Processes, Cognitive model, Orchestration

ACM Reference Format:
Andrea Loreggia, Nicholas Mattei, Taher Rahgooy, Francesca Rossi, Biplav Srivastava, and Kristen Brent Venable. 2022. Making Human-Like Moral Decisions. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society (AIES '22), August 1–3, 2022, Oxford, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3514094.3534174

1 INTRODUCTION

Implicit and explicit constraints are present in many decision making scenarios, and they force us to make difficult decisions: do we always satisfy all constraints, or do we violate some of them in exceptional circumstances? These decisions are especially difficult when such constraints model ethics or safety-related principles and policies, since violating such constraints may be perceived as our inability to live by our moral principles or to take care of our safety. We believe that machines can help humans improve their moral and safety decisions, by alerting them when they predict that they will probably make unethical moves in their decision environment [2–4, 9, 10, 14, 16]. The first step to achieve this vision is to build machines that are able to make these predictions with high accuracy. In this paper, we propose an approach to build an AI agent that can accurately mimick human decision making in an ethically constrained environment.

We model our environment as a Markov Decision Process (MDP). In particular, we consider a transition system with states and actions that allow an agent to transition from one state to another one. Each task has an initial state and a goal state, to be reached through a sequence of actions while minimizing the length of the sequence and the number of constraint violations. The constraints we consider model ethical and safety rules and policies in a comprehensive way, by including both constraints on actions, that model deontological rules such as "don't kill" or "don't drive too fast", and constraints over states, that model consequentialist policies such as "any action is possible, but make sure you don't end up in a certain situation". Moreover, state constraints can refer to specific states (such as in "make sure to avoid having a too high blood pressure") or also classes of states (such as in "avoid any situation where people are in danger"). In this constrained decision environment, we build an AI agent that finds trade-offs between reaching the goal with the shortest sequence of moves and satisfying the constraints.

Many techniques can be used to build AI agents that rationally minimize constraint violations while achieving a given goal [12, 20]. However, here the aim is to mimic human behavior, and it is well known that humans are not rational, especially when confronted with decisions that require making trade-offs between collective norms and personal objectives, where they often reason by employing heuristics and approximations which are subject to bias and noise [4, 5]. Therefore, techniques that employ optimal rationality are not suitable to mimick human behavior [14]. For this reason, our AI agent is inspired by a cognitive theory of human deliberation, called "multi-alternative decision field theory" (MDFT) [5]. MDFT is a psychological theory of how humans make decisions that has been shown to be able to capture deviations from rationality observed in humans, making trade-offs between competing objectives in a human-like way. Here we use MDFT to decide on each individual action in the sequence from the initial to the goal state, providing all the possible actions (both constrained and not) at each step. MDFT also includes a way to focus attention on individual features of the options. We use this to allow our AI agent to focus on either reaching the goal state or satisfying the constraints. Given that our AI agent effectively orchestrates between these two competing desires, we call it MDFT-O (for MDFT-Orchestrator).

We study MDFT-O both theoretically and experimentally, showing that our architecture is theoretically more expressive and obtains better empirical performance compared to other orchestrators across a range of metrics. We also compare the action sequences generated by MDFT-O with those generated by humans and obtained via data collected through Amazon Mechanical Turk (AMT), showing that MDFT-O generates human-like trajectories much better than other orchestrators.

Summarizing, our contribution in this paper are as follows:

- We define a constrained decision environment which models both deontological and consequentialist ethics policies;
- We build an AI orchestrator agent that acts in this decision environment by making trade-offs between reaching the goal and satisfying the constraints;
- We prove theoretically that our agent is strictly more general than other orchestrators;
- We also prove experimentally that it generates better action trajectories than other orchestrators, and that it can faithfully mimic human trajectories.

2 MARKOV DECISION PROCESSES AND CONSTRAINTS

In order to define and test MDFT-O, we use Markov Decision Processes (MDPs) to model the decision environment where actions involving trade-offs take place. MDPs are a general model of decision making widely used in artificial intelligence and robotics [19]. Formally, a finite-horizon Markov Decision Process (MDP) $\mathcal{M}$ is a model for sequential decision making over time steps $t \in T$ is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, D_0, \phi, \gamma, R)$ [19] where:

- $\mathcal{S}$ is a finite set of discrete states;
- $\{\mathcal{A}_s\} \subseteq \mathcal{A}$ is a set of actions available at state s;
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a model of the environment given as transition probabilities where $P(s_{t+1}|s_t, a_t)$ is the probability of transitioning to state s_{t+1} from state s_t after taking action $a_t \in \{\mathcal{A}_{s_t}\}$ at time t;
- $D_0 : \mathcal{S} \to [0, 1]$ is a distribution over start states;
- $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^k$ is a mapping from the transitions to a k-dimensional space of features; $\gamma \in [0, 1)$ is a discount factor; and
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a scalar reward received by the agent for being in one state (s_t) and transitioning to another state (s_{t+1}) at time t, written as $R(s_t, a_t, s_{t+1})$.

Within the environment defined by the MDP, agents generate a sequence of actions called a *trajectory* of length t. Let $\tau = ((s_1, a_1, s_2), ..., (s_{t-1}, a_{t-1}, s_t)) \in (\mathcal{S} \times \mathcal{A} \times \mathcal{S})^t$. We evaluate the quality of a particular trajectory in terms of the amount of reward accrued over the trajectory, subject to discounting, $R(\tau) = \sum_{i=1}^{t} \gamma^i R(s_i, a_i, s_{i+1})$. A policy, $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a map of probability distribution to actions for every state such that $\pi(s, a)$ is the probability of taking action a in state s. The probability of a trajectory τ under a policy is $\pi(\tau)$. The goal is to find π^* that maximizes the expected reward, $J(\pi) = \mathbb{E}_{\tau\sim\pi}[R(\tau)]$ [11]. Classical tabular methods can be used to find π^*, e.g, value iteration (VI). Such a method finds an optimal policy by estimating the expected reward for taking an action a in a given state s, i.e., the Q-value of pair (s, a), written $q(s, a)$ [19].

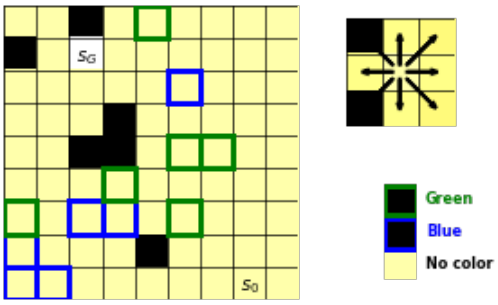

Figure 1: Example environment for our agents. The agent must move from S_0 to S_G while not violating various constraints (black squares). Constraints can be over actions (top right, black squares), state features (green and blue color, bottom right), or state occupancy (black squares in the grid, left). Violating a constraint incurs a penalty.

A Constrained MDP $\mathcal{M}^C$ is a nominal MDP $\mathcal{M}^N$ with an additional cost function $C : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and a budget $\alpha \geq 0$. We can then define the cost of a trajectory to be $c(\tau) = \sum_{i=1}^{t} c(s_i, a_i, s_{i+1})$ [1, 11]. Setting $\alpha = 0$ is enforcing *hard constraints*, i.e., we must never trigger constrained transitions. Under a soft constraints paradigm, each constraint comes with a real-valued penalty/cost and the goal is to minimize the sum of penalties incurred by the agent. We consider three very general types of constraints, illustrated in Figure 1, which could arise from ethical or moral considerations:

Action Constraints: An agent should not perform some (set of) action a_i;

Occupancy Constraints: An agent should not occupy a (set of) states s_i;

Feature Constraints: Given a feature mapping of transitions ϕ, an agent should not perform an (set of) action in presence of specific state features.

For example, the task of driving to a destination under time pressure, the nominal world $\mathcal{M}^N$ would correspond to the unconstrained environment, where we can reach our goal along any path, regardless of its safety or compliance with traffic regulations. On the other hand, in the constrained world $\mathcal{M}^C$ we would have action constraints preventing wrong-way driving, occupancy constraints discouraging driving on sidewalks, and feature constraints deterring from using any lane reserved for public transport.

The optimal policies for $\mathcal{M}^N$ and $\mathcal{M}^C$, denoted with π_n and π_c, can, thus, be seen as representing optimal strategies in terms reaching the destination as quickly as possible and driving safely, respectively.

3 MULTI-ALTERNATIVE DECISION FIELD THEORY (MDFT)

In this paper, we propose using a cognitive model of decision making to orchestrate between pursuing goals and satisfying constraints. Decision field theory (DFT) is a dynamic-cognitive approach that models human decision making based on psychological principles [5]. DFT models the preferential choice as an accumulative process in which the decision maker attends to a specific attribute at each time to derive comparisons among options and update his preferences accordingly. Ultimately the accumulation of those preferences forms the decision maker's choice. DFT has been extended by [15] to multialternative preferential choice (denoted MDFT, for Multialternative DFT), where an agent is confronted with multiple options and equipped with an initial personal evaluation for them according to different criteria called attributes. For example, a student who needs to choose a main course among those offered by the cafeteria will have in mind an initial evaluation of the options in terms of how tasty and healthy they look. More formally, MDFT, in its basic formulation [15], is composed of:

Personal Evaluation: We assume a set of options $\{o_1, \ldots, o_n\}$ and a set of attributes $\{A_1, \ldots, A_J\}$. The subjective value of option o_i on attribute A_j is denoted by m_{ij} and stored in matrix $\mathbf{M}$ for all options and attributes. In our example, let us assume that the cafeteria options for main course are *Salad (S)*, *Burrito (B)* and *Vegetable pasta (V)* and that the attributes considered are *Taste* and *Health*. Matrix $\mathbf{M}$ containing the student's initial preferences for the three options according to the two attributes could be defined as follows:

$$\mathbf{M} = \begin{bmatrix} 1 & 5 \\ 5 & 1 \\ 2 & 3 \end{bmatrix}$$

In this matrix the rows correspond to the options in order (S, B, V) and the columns to the attributes $Taste$ and $Health$. For example, we can see that $Burrito$ has a high preference in terms of taste but low in terms of nutritional value.

Attention Weights: Attention weights are used to express how much attention is allocated to each attribute at each particular time t during the deliberation process. We denote them by a one-hot column vector $\mathbf{W}(t)$ where $W_j(t)$ is a value denoting the attention to attribute j at time t. We adopt the common simplifying assumption that, at each point partial, the decision maker attends to only one attribute. Thus, $W_j(t) \in \{0, 1\}, \forall t, j$. In our example, where we have two attributes, at any point in time t, we will have $\mathbf{W}(t) = [1, 0]$, or $\mathbf{W}(t) = [1, 0]$, representing that the student is attending to, respectively, $Taste$ or $Health$. In general, the attention weights change across time according to a stationary stochastic process with probability distribution $\mathbf{w}$, where w_j is the probability of attending to attribute A_j. In our example, defining $w_1 = 0.55$ and $w_2 = 0.45$ would mean that at each point in time, the student will be attending $Taste$ with probability 0.55 and $Health$ with probability 0.45. In other words, $Taste$ matters slightly more to this particular student than $Health$.

Contrast Matrix: Contrast matrix $\mathbf{C}$ is used to compute the advantage (or disadvantage) of an option with respect to the other options. For example, $\mathbf{C}$ can be defined by contrasting the initial evaluation of one alternative against the average of the evaluations of the others. In this case, for three options, we have:

$$\mathbf{C} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}$$

At any moment in time, each alternative in the choice set is associated with a **valence** value. The valence for option o_i at time t, denoted $v_i(t)$, represents its momentary advantage (or disadvantage) when compared with other options on some attribute under consideration. The valence vector for n options $o_1, \ldots, o_n$ at time t, denoted by column vector $\mathbf{V}(t) = [v_1(t), \ldots, v_n(t)]^T$, is formed by:

$$\mathbf{V}(t) = \mathbf{C} \times \mathbf{M} \times \mathbf{W}(t) \tag{1}$$

In our example, the valence vector at any time point in which $\mathbf{W}(t) = [1, 0]$, is $\mathbf{V}(t) = [1 - 7/2, 5 - 3/2, 2 - 6/2]^T$.

Preferences for each option are accumulated across the iterations of the deliberation process until a decision is made. This is done by using **Feedback Matrix** $\mathbf{S}$, which defines how the accumulated preferences affect the preferences computed at the next iteration. This interaction depends on how similar the options are in terms of their initial evaluation expressed in $\mathbf{M}$. Intuitively, the new preference of an option is affected positively and strongly by the preference it had accumulated so far, while it is inhibited by the preference of similar options. This lateral inhibition decreases as the dissimilarity between options increases. In our example, by following the standard method of defining the $\mathbf{S}$ matrix described in [8], we obtain $\mathbf{S}$ matrix:

$$\mathbf{S} = \begin{bmatrix} +0.9000 & -0.0000 & -0.0405 \\ -0.0000 & +0.9000 & -0.0047 \\ -0.0405 & -0.0047 & +0.9000 \end{bmatrix}$$

At any moment in time, the preference of each alternative is calculated by

$$\mathbf{P}(t+1) = \mathbf{S} \times \mathbf{P}(t) + \mathbf{V}(t+1) \tag{2}$$

where $\mathbf{S} \times \mathbf{P}(t)$ is the contribution of the past preferences and $\mathbf{V}(t+1)$ it the valence computed at that iteration. Usually the initial state $\mathbf{P}(0)$ is defined as $\mathbf{0}$, unless defined otherwise due, for example, to prior knowledge on past experiences.

Given an MDFT model, one can simulate the process of deliberating among the options by accumulating the preferences for a number of iterations. The process can be stopped either by setting

a threshold on the preference value and selecting whichever option reaches it first or, by fixing the number of iterations and then selecting the option with highest preference at that point. In general, different runs of the same MDFT model may return different choices because of the uncertainty on the attention weights distribution. In this way, MDFT induces choice distributions over set of options and is capable of capturing well know behavioral effects such as the compromise, similarity, and attraction effects that have been observed in humans and that violate rationality principles [6].

4 ORCHESTRATING GOALS AND CONSTRAINTS

Often humans are confronted with decisions that require making trade-offs between collective norms and personal objectives [12, 17]. In this section, we investigate different methods for combining policies π_n for the nominal $\mathcal{M}^N$ and π_c for the constrained $\mathcal{M}^C$.

For every state action pair (s, a), we consider vectors $\langle sq_n(s,a)\rangle$ and $\langle sq_c(s,a)\rangle$ with $i \in \{1,\dots,k\}$ where $sq_n(s,a)$ (resp. $sq_c(s,a)$) represents the probability of choosing action a in state s according to policy π_n (resp. π_c). For example, if π_n and π_c are learned using VI or Q-learning, then, such probabilities are obtained by taking the softmax of the Q-values for each policy. We define the following orchestrating policies:

Greedy Orchestrator G-O: uses policy π_G, where $\pi_G(s) = a$, selects an action a with overall highest sq-value:

$$a = \underset{a\in\mathcal{A}_s}{argmax\,max}\{sq_c(s,a), sq_n(s,a)\}.$$

Weighted Average Orchestrator WA-O: is defined by policy π_{WA}. Given weight vector (w_n, w_c) with w_n,$w_c \in [0,1]$ and $w_c + w_n = 1$, action $a = \pi_{WA}(s)$ is chosen according to probability distribution $p_{WA}(a_i) = w_n sq_n(s,a_i) + w_c sq_c(s,a_i)$.

MDFT Orchestrator MDFT-O: chooses actions according to policy π_{MDFT}. Action $a = \pi_{MDFT}(s)$ is chosen via an MDFT model where: $\mathbf{M}$ is a $k \times 2$ matrix where rows (i.e., options) correspond to actions and columns (i.e., attributes) correspond to $\mathcal{M}^N$ and $\mathcal{M}^C$. The i-th element of the respective world column is $sq_n(s,a_i)$ (resp., $sq_c(s,a_i)$), i.e., we are using the probability of choosing an action as a proxy of its preference. The weight vector (w_n, w_c) is defined as for π_{WA}, and serves as probability distribution $\mathbf{w}$ defining how attention shifts between attributes during deliberation. Matrices $\mathbf{C}$ and $\mathbf{S}$ are defined in the standard way as described in Section 3. When reaching state s, an MDFT deliberation process is launched to decide which action should be chosen. At each step the focus is shifted to $\mathcal{M}^N$ or $\mathcal{M}^C$ according to probability distribution (w_n, w_c), and the preferences of the actions according to the selected attribute are accumulated as per Section 3.

Informally, G-O is a deterministic approach that takes the most promising action, WA-O allows the agent to compromise between the pursuit of the goal state and satisfying constraints via a new policy obtained by considering the weighted average of the nominal and constrained distributions, and the MDFT-based orchestrator, MDFT-O, uses MDFT to chose at each step an action using a psychology-grounded simulation of how humans deliberate.

5 THEORETICAL COMPARISON OF ORCHESTRATION METHODS

We first compare theoretically the expressive power of the three orchestrators, G-O, Weighted WA-O and MDFT-O. We focus on a single state s and consider how the policies compare in terms of being able to model a given distribution over the actions available in s. We start by considering G-O that is deterministic and will pick a fixed action a in state s. Both WA-O and MDFT-O can model the Greedy policy by shifting all the weight to the environment where the maximum value is obtained and zeroing all preferences except for that of action a. More formally:

Theorem 1. *Consider state s. Any choice probability distribution over the actions available in s that can be WA-O approaches.*

Proof. We can model the (degenerate) probability distribution induced by G-O via an MDFT with as many options as the actions available in s, two attributes with weights set to any random pair of values, and preferences in the $\mathbf{M}$ matrix all equally to 0 except for those in the row associated with a which are set to 1. Matrices $\mathbf{C}$ and $\mathbf{S}$ can be defined in the standard way described in Section 3 and deliberation can be halted after one deliberation step. In fact, when deliberation is launched, an attribute will be selected. Regardless of which one is selected, action a will be chosen given that it is the only one with non-zero preference.

Similarly, we can model the G-O distribution using a WA-O where $w_n = w_c = 1/2$, and $sq_n(s,a) = sq_c(s,a) = 1$ and $sq_n(s,a') = sq_c(s,a') = 0$, $\forall a' \neq a$. □

This observation, along with the fact that MDFT-O and WA-O are non-deterministic, allows us to conclude that G-O is strictly less expressive than the other two orchestrators.

Turning to the comparison between MDFT-O and WA-O, we can prove the following statement.

Theorem 2. *Given any state s, there exist choice probability distributions over the actions available in s that can be modeled by MDFT-O but not by WA-O.*

Proof. We use an instance of the well known compromise effect [5] according to which a compromising alternative tends to be chosen more often by humans than options with complementary preferences with respect to the attributes. Consider the case of state s with three actions a_1, a_2 and a_3. Let us assume that, for example, $sq_n(s,a_1) = 1/6$, $sq_n(s,a_2) = 1/3$ $sq_n(s,a_3) = 1/2$ and $sq_c(s,a_1) = 1/2$, $sq_c(s,a_2) = 1/3$ $sq_c(s,a_3) = 1/6$. According to the compromise effect humans will tend to choose a_2 more often than a_1 and a_3. Such a choice distribution over the actions can be modeled by an MDFT defined over option set $\{a_1, a_2, a_3\}$, with two attributes and weights $w_n = 0.55$ and $w_c = 0.45$ [5]. However, if we now consider WA-O, we can see that there is no way to define weights (w_n, w_c) such that the corresponding weighted average probability satisfies $w_n sq_n(s,a_2) + w_c sq_c(s,a_2) > max\{w_n sq_n(s,a_1) + w_c sq_c(s,a_1), w_n sq_n(s,a_3) + w_c sq_c(s,a_3)\}$. Thus, this distribution over actions cannot be modeled by the WA-O. □

On the other hand, if we consider MDFTs in general, i.e. without the restriction of having two attributes, we can model any distribution.

Theorem 3. *Given state s and the set $\mathcal{A}_s$ of actions available in s, consider a probability distribution p defined over $\mathcal{A}_s$. We can define an MDFT model where the set of options corresponds to $\mathcal{A}_s$ and the induced choice probability distribution coincides with p.*

Proof. Consider the MDFT model defined as follows:

- Matrix $\mathbf{M}$ is the $k \times k$ identity matrix;
- Weight vectors $\mathbf{W}$ are defined as in Section 3 and select a single attribute at each iteration. Probability distribution over attributes $\mathbf{w}$ is defined in a way such that the probability of selecting the j-th attribute, is $p(a_j)$.
- Matrices $\mathbf{C}$ and $\mathbf{S}$ are defined in the standard way as described in Section 3.
- The deliberation time for the model is fixed at one iteration.

It is easy to see that running the model induces a choice probability over the actions which corresponds to p. In fact, in every run, which consists of a single iteration, an attribute A_h will be sampled according to probability p. Given how $\mathbf{M}$ is defined and the fact that the initial value of the accumulated preference $\mathbf{P}(0) = 0$, action a_h will be chosen. Thus, the probability of action a_h being selected, given the MDFT model, coincides with $p(a_h)$. □

As a consequence, MDFT-O is general enough to express the probability distributions induced over the actions by WA-O.

Whether this is true also in the case of MDFTs with only two attributes, as used by MDFT-O, remains an open theoretical question. However, we verify this experimentally. In Rahgooy and Venable [13], the authors propose an RNN-based approach that starts from samples of a choice distribution and recovers parameters of an MDFT model, minimizing the divergence between the original and MDFT-induced choice distributions. We adapt their code[1] and generate 100 instances of WA-O distributions starting from random sq_n and sq_c distributions and (w_n, w_c) weights. For each of these instances we generate 100 samples (i.e, chosen actions). We fix the sq_n and sq_c values as parameters for the $\mathbf{M}$ matrix and learn the attention weight distribution $\mathbf{w}$ using 300 learning iterations. We use the learned MDFT model to generate a choice distribution over the actions with a stopping criteria of 25 deliberations steps. The observed average JS divergence between the original WA-O distributions and the ones induced by learned MDFT is 0.024 with standard error 0.0013. This shows experimentally that we can learn weights for an MDFT model to replicate any choice distribution of WA-O.

6 EXPERIMENTAL COMPARISON OF ORCHESTRATION METHODS

We now compare the G-O, WA-O, and MDFT-O orchestrators empirically to test if the orchestration techniques can be leveraged to create agents that trade-off between conflicting objectives like humans. We first compare the orchestrators on synthetic data and then collect decision making data from humans and compare the orchestrators on how well they mimic the choices of human decision makers.

[1] Available at https://github.com/Rahgooy/MDFT

6.1 Synthetic Experiments

For our synthetic experiments, we adopt a similar gridworld setup that has been used in many recent studies of decision making in constrained MPDs [7, 18]. An example of navigating in this gridworld task is depicted in Figure 2.

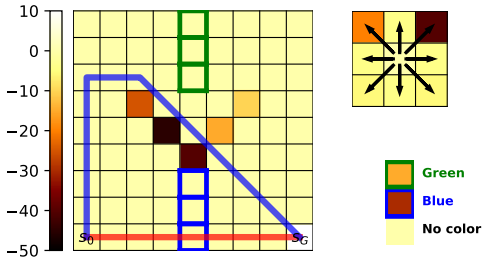

Figure 2: Example environment for the synthetic experiments. The red trajectory depicts an agent that goes directly to the goal and ignores any constraints while the blue line attempts to navigate to the goal while still respecting the constraints.

For both $\mathcal{M}^N$ and $\mathcal{M}^C$, we adopt a similar gridworld setup as Scobee and Sastry [18] (see Figure 2): we set an action penalty of -4 for the cardinal directions, $-4 \times \sqrt{2}$ for taking the diagonal, and a reward of 10 for reaching the goal state. For $\mathcal{M}^C$, we also fix the constraint costs on the generated grids for states, actions, and features to be -50. Throughout, we assume non-deterministic worlds with a 10% chance of action failure, resulting in a random action.

We start by generating 100 different non-deterministic nominal worlds, $\mathcal{M}^N$ and corresponding constrained worlds. We learn, via value iteration on both $\mathcal{M}^N$ and $\mathcal{M}^C$, the optimal policy for that world denoted π_n and π_c, respectively, along with the associated q values for each state under the optimal policy: q_n and q_c.

Both the MDFT-O and WA-O agents can vary the weight that each places on q_n and q_c. Hence to compare them, and in each of our tests, we sweep these weighting values from $(0, 1)$ in steps of 0.1. This gives us a pair (n, c) where the value for n means that more weight is placed on the q_n values and the value for c is the weight for the q_c values. To avoid issues with the differing scales of rewards, we first apply a softmax to the q values before combining them as applying the softmax forces the q values to a probability distribution that is comparable.

For all our results, we generated 200 trajectories for each step and method (including π_n and π_c, denoted as Nominal and Constrained in Fig. 3), and for each of the 100 random worlds. For each world, we compute the probability distribution over the transitions (s_t, a_t, s_{t+1}) counting the number of times a transition occurs in a generated trajectory. We first test to ensure that the trajectories generated by MDFT-O and WA-O are statistically significantly different. To do this, we perform a Kolmogorov-Smirnov test and confirm that the two techniques induce statistically significantly different

choice distributions, rejecting the null at every weight step with $p \leq 0.01$.

In what follows, we normalize the values in order to have comparable plotted values; the results of our experiments are shown in Figure 3. For comparison, we ran a q-learning agent to find the optimal policy for both the nominal world and the learned constrained world. These are shown in our results as the red and blue dashed lines. Note that at $(1, 0)$ (resp. $(0, 1)$) WA-O is equivalent to π_n (resp. π_c), and that in both cases MDFT-O becomes deterministic, picking the action with highest Q-value.

Figure 3 (top left) shows the average length of trajectories produced by the orchestrators, normalized so that 1.0 is the shortest path between the start and goal state; (top right) the average normalized penalty for trajectories in $\mathcal{M}^C$, lower is better; (bottom left) the average number of violated constraints. Across all these metrics, MDFT-O is performing better than WA-O by always reaching the goal in a smaller number of steps no matter the configuration of the orchestrator. We can also see that MDFT-O agent violates fewer constraints and accumulates lower penalties.

Finally, in Figure 3 (bottom right) we show the JS Divergence between the trajectories generated by $\pi^*_{C^*}$ and the trajectories generated by MDFT and WA, as the weight vector varies. Given two sets of trajectories, we compute the Jensen–Shannon (JS) divergence between the two distributions induced by the policies: $div_{js}(p, q) = {}^{(D(p||m)+D(q||m))}/_2$, where m is the pointwise mean of p and q and D is the Kullback-Leibler divergence. This metric allows us to quantify the similarity between the two distributions and thus to define the similarity between the orchestrator and the demonstrations.

For both agents, the divergence is small on the left and grows moving to the right, as constraints become less important. This is not surprising, since the reference trajectories are generated using $\pi^*_{C^*}$. Furthermore, we note that the MDFT advantage is more significant when w_c is larger, that is when constraints matter more. An explanation for this is that a large value w_c results in more MDFT deliberation steps to be focused (exclusively) on preferences relative to the constrained world. In WA, the averaging of the values underlying the policies, although weighted, is not able to maintain the importance of the constraints.

6.2 Human Experiments

We conducted an Amazon Mechanical Turk study to get inputs on how humans may navigate the grid. For this purpose, questions were posed for every cell in the grid shown in Figure 1. Each question corresponded to the decision an agent may take at that location. The questions were framed in terms of choosing different roads at an intersection where each road is labeled with two values: the first one representing how fast (but possibly unsafely) it will take the respondent to destination and the other one representing how safe, but possibly slower, the road is. The roads were used as proxies of the actions on the grid and the scores presented to the participants were obtained by multiplying the corresponding sq_n and sq_c values by 100. An example of a question is shown in Figure 4.

For the complete 9x9 grid, there were 81 questions with a question having a maximum of 8 solution choices (questions regarding boundary cells had fewer). The questions were divided into 9 sub-surveys consisting of 9 questions and a validation question (requiring the sub-survey number to be entered). The participants were given a common survey link and a participant could be assigned any of the 9 sub-surveys [2]. A total of 185 participants responded and we obtained an average of 21 responses per cell. We then used the frequencies with which participants chose the different roads to obtain a probability distribution over the actions of the corresponding cell. Finally we generated trajectories by starting from the initial state (at the bottom right of the grid depicted in Figure 1) and repeatedly sampled actions according to the obtained distributions until we reached the goal. In what follows, we will refer to these trajectories as human trajectories. In Figure 5, we show the JS divergence between 200 trajectories generated by using respectively WA-O and MDFT-O with different settings of the attention weights and 200 human trajectories. As it can be seen, MDFT-O outperforms WA-O for every combination of weights thus confirming its superiority in capturing human decision making even in this complex setting with multiple options.

Moreover, we can see that MDFT-O trajectories generated are closest to the human ones when attention weights are set to 0.2 for the nominal world and 0.8 for the constrained world (minimal obtained JS-divergence is 0.262337). This suggests that, on average, participants cared substantially more about safe driving than reaching the destination at all costs. In Figure 6, we show a graphical comparison of the human trajectories and the MDFT-O trajectories for the optimal attention weights to further illustrate their similarity.

Summarizing, our results suggest that MDFT-O is a promising tool for modeling how humans trade off between pursuing objectives and minimizing constraint violations. Moreover, the model can be used to fit behavioral data and to predict the priorities underlying the orchestration.

7 CONCLUSIONS AND FUTURE WORK

We defined a constrained decision environment which models both deontological and consequentialist ethical constraints, where the considered task is to reach a goal state with the shortest path while satisfying the ethical constraints. We have built an AI orchestrator agent that acts in this decision environment by making trade-offs between reaching the goal and satisfying the constraints, proving that it is more general than other orchestrators and generates better trajectories. We have also shown that our AI agent faithfully mimicks human trajectories, thus providing a way to predict how human would make decisions and possibly help them improve their decision quality and satisfaction.

8 ACKNOWLEDGEMENTS

We will like to thank Sai Teja Paladi for his help in implementing randomization mechanisms for AMT survey. Nicholas Mattei was supported by NSF Awards IIS-RI-2007955, IIS-III-2107505, and IIS-RI-2134857, as well as an IBM Faculty Award and a Google Research

[2] **IRB Exemption and Compensation**. This research study has been certified as exempt from the IRB per 45 CFR 46.104(d)(3) and 45 CFR 46.111(a)(7) by University of South Carolina IRB# Pro00118795. Participants were compensated at a rate of 1 USD per survey attempt consisting of 9 questions and the validation check. Amazon charged an additional administrative fee of 40%.

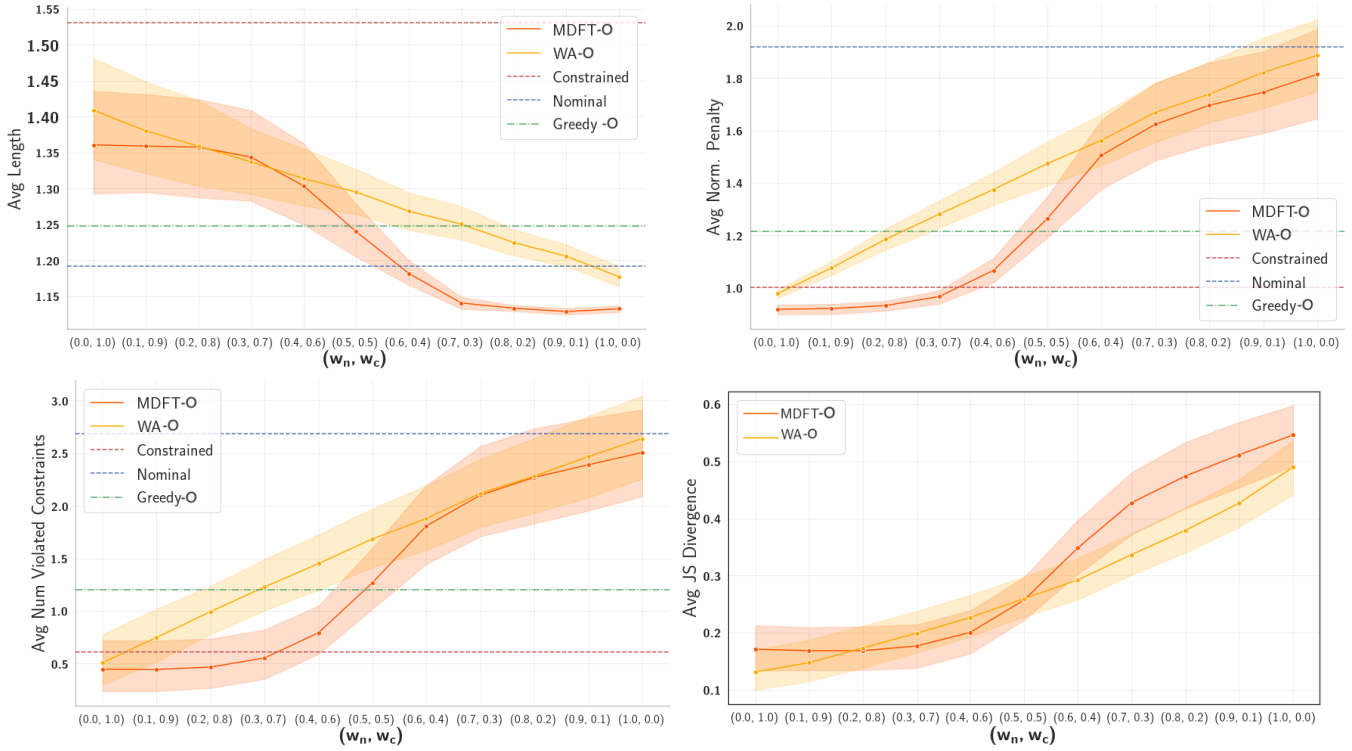

Figure 3: Comparison of Greedy, WA, and MDFT on our synthetic datasets in terms of average path length (top left), normalized penalty (lower is better, top right), average number of violated constraints (lower is better, bottom left), and average JS Divergence between π_c trajectories and orchestrators MDFT and WA (bottom right).

6) You are driving your car and arrive at an intersection, from where you can proceed along several roads. For each road, we provide you with two pieces of information:

Quick but Unsafe: The first number is a score between 0 and 100 representing how good the road is in terms of getting you to your goal quickly, but perhaps unsafely.

Safe but Delay: The second number is a score between 0 and 100 representing how good the road is in terms of getting you to your goal safely, but perhaps with delays on the way. *

- Road 1: Quick but Unsafe score: 67, Safe but Delay score: 0
- Road 2: Quick but Unsafe score: 0, Safe but Delay score: 0
- Road 4: Quick but Unsafe score: 2, Safe but Delay score: 92
- Road 7: Quick but Unsafe score: 26, Safe but Delay score: 0
- Road 8: Quick but Unsafe score: 0, Safe but Delay score: 0

Figure 4: Example of survey question with five options.

Scholar Award. Biplav Srivastava is funded by a University of South Carolina innovation grant and VAJRA award from Govt. of India.

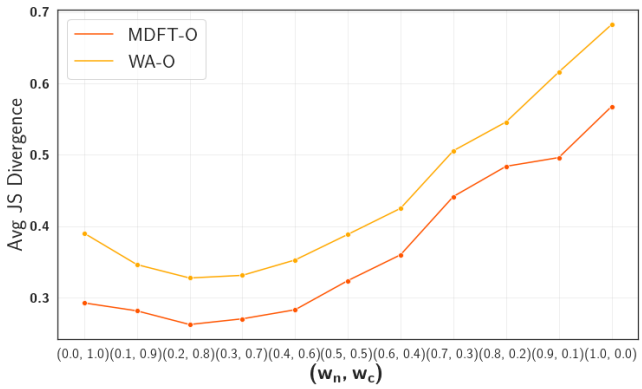

Figure 5: Average JS Divergence between trajectories generated from the human subjects experiment and MDFT-O and WA-O.

REFERENCES

[1] Eitan Altman. 1999. *Constrained Markov Decision Processes*. Vol. 7. CRC Press.
[2] A. Balakrishnan, D. Bouneffouf, N. Mattei, and F. Rossi. 2018. Using Contextual Bandits with Behavioral Constraints for Constrained Online Movie Recommendation. In *Proc. of the 27th Intl. Joint Conference on AI (IJCAI)*.
[3] Avinash Balakrishnan, Djallel Bouneffouf, Nicholas Mattei, and Francesca Rossi. 2019. Incorporating Behavioral Constraints in Online AI Systems. In *Proc. of*

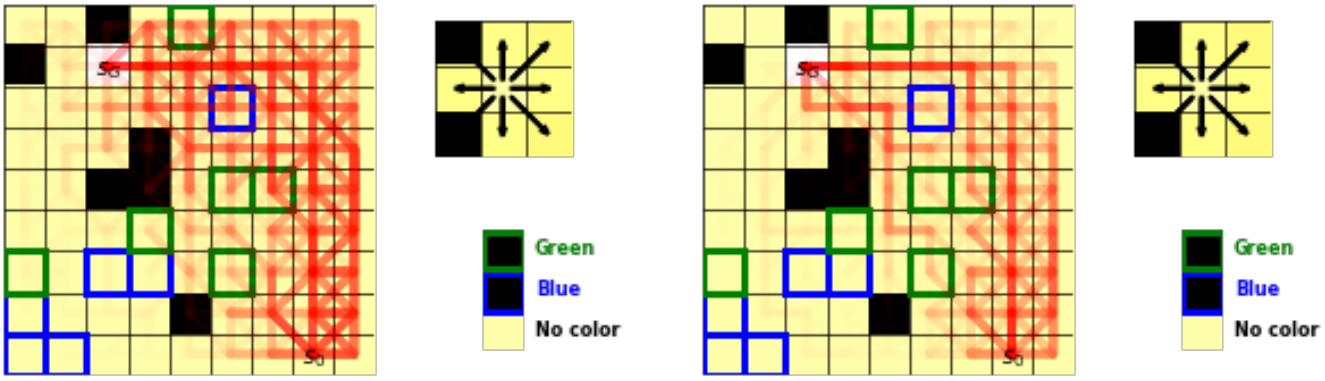

Figure 6: Illustration of 200 human trajectories (left) and 200 MDFT-O trajectories (right) generated with attention weight vector (0.2, 0.8). Initial and goal states are labeled with S_0 and S_G, respectively, and transitions are depicted in red with thickness representing the frequency with which they appeared in a trajectory.

the 33rd AAAI Conference on Artificial Intelligence (AAAI).

[4] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andrea Loreggia, Keerthiram Murugesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. 2021. Thinking Fast and Slow in AI. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 15042–15046. https://ojs.aaai.org/index.php/AAAI/article/view/17765

[5] Jerome R Busemeyer and Adele Diederich. 2002. Survey of decision field theory. *Mathematical Social Sciences* 43, 3 (2002), 345–370.

[6] Jerome R Busemeyer and James T Townsend. 1993. Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. *Psychological review* 100, 3 (1993), 432.

[7] Arie Glazier, Andrea Loreggia, Nicholas Mattei, Taher Rahgooy, Francesca Rossi, and Brent Venable. 2022. Learning Behavioral Soft Constraints from Demonstrations. arXiv:2202.10407 [cs.LG]

[8] Jared M Hotaling, Jerome R Busemeyer, and Jiyun Li. 2010. Theoretical developments in decision field theory: Comment on Tsetsos, Usher, and Chater (2010). *Psychological Review* 117, 4 (2010).

[9] Andrea Loreggia, Nicholas Mattei, Francesca Rossi, and K Brent Venable. 2019. CPM etric: Deep Siamese Networks for Metric Learning on Structured Preferences. In *International Joint Conference on Artificial Intelligence*. Springer, 217–234.

[10] Andrea Loreggia, Nicholas Mattei, Francesca Rossi, and K Brent Venable. 2020. Modeling and reasoning with preferences and ethical priorities in AI systems. *Ethics of Artificial Intelligence* 127 (2020).

[11] Shehryar Malik, Usman Anwar, Alireza Aghasi, and Ali Ahmed. 2021. Inverse Constrained Reinforcement Learning. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 7390–7399. https://proceedings.mlr.press/v139/malik21a.html

[12] Ritesh Noothigattu, Djallel Bouneffouf, Nicholas Mattei, Rachita Chandra, Piyush Madan, Kush R. Varshney, Murray Campbell, Moninder Singh, and Francesca Rossi. 2019. Teaching AI agents ethical values using reinforcement learning and policy orchestration. *IBM J. Res. Dev.* 63, 4/5 (2019), 2:1–2:9. https://doi.org/10.1147/JRD.2019.2940428

[13] Taher Rahgooy and K. Brent Venable. 2019. Learning Preferences in a Cognitive Decision Model. In *Human Brain and Artificial Intelligence*, An Zeng, Dan Pan, Tianyong Hao, Daoqiang Zhang, Yiyu Shi, and Xiaowei Song (Eds.). Springer Singapore, Singapore, 181–194.

[14] Mark O Riedl. 2019. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies* 1, 1 (2019), 33–36.

[15] Robert M Roe, Jermone R Busemeyer, and James T Townsend. 2001. Multialternative decision field theory: A dynamic connectionst model of decision making. *Psychological review* 108, 2 (2001), 370.

[16] Francesca Rossi and Andrea Loreggia. 2019. Preferences and Ethical Priorities: Thinking Fast and Slow in AI.. In *AAMAS*. 3–4.

[17] F. Rossi and N. Mattei. 2019. Building Ethically Bounded AI. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

[18] Dexter R. R. Scobee and S. Shankar Sastry. 2020. Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=BJliakStvH

[19] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction, 2nd Edition*. A Bradford Book, Cambridge, MA, USA.

[20] Justin Svegliato, Samer B Nashed, and Shlomo Zilberstein. 2021. Ethically compliant sequential decision making. In *Proceedings of the 35th AAAI International Conference on Artificial Intelligence (AAAI)*.