Environment Design for Biased Decision Makers

Guanghui Yu and **Chien-Ju Ho**
Washington University in St. Louis
{guanghuiyu, chienju.ho}@wustl.edu

Abstract

We study the *environment design* problem for biased decision makers. In an environment design problem, an informed *principal* aims to update the decision making environment to influence the decisions made by the *agent*. This problem is ubiquitous in various domains, e.g., a social networking platform might want to update its website to encourage more user engagement. In this work, we focus on the scenario in which the agent might exhibit biases in decision making. We relax the common assumption that the agent is rational and aim to incorporate models of biased agents in environment design. We formulate the environment design problem under the Markov decision process (MDP) and incorporate common models of biased agents through introducing general time-discounting functions. We then formalize the environment design problem as constrained optimization problems and propose corresponding algorithms. We conduct both simulations and real human-subject experiments with workers recruited from Amazon Mechanical Turk to evaluate our proposed algorithms.

1 Introduction

We explore the problem where two parties with mis-aligned objectives, a *principal* and an *agent*, are in the same sequential decision making environment. The goal of the agent is to take a sequence of actions to maximize his total payoff[1]. The principal cannot directly take actions but can update the environment to influence the agent's actions and receive reward based on the agent's actions. The goal of the principal is to update the environment such that the agent takes actions that maximize the principal's payoff.

This problem setting is motivated by several existing and potential applications. For example, a user-generated content website might want to update their site to provide incentives, such as badges or virtual points, to encourage users to consume and rate the content on their website. An online retailer might want to decide when and whether to provide coupons to nudge the user to make the purchase. An assistive AI agent might want to provide interventions, such as reminding messages, to help humans achieve personal goals, such as reducing the amount of time spent on social networking sites.

If we assume the agent is *rational* and makes decisions according to the optimal policy, this problem is similar to several existing works in the literature, including policy teaching [Zhang and Parkes, 2008; Zhang *et al.*, 2018], in which the principal updates the reward functions to induce the agent to take certain policies, and the poisoning attack for reinforcement learning [Rakhsha *et al.*, 2020; Zhang *et al.*, 2020], in which an adversarial principal aims to modify the training environment such that the agent learns the undesired policy. In this work, we are motivated by the natural setting in which the agent is a human being and might exhibit biases in decision making. As observed in empirical studies, humans are known to exhibit systematic biases in making decisions. For example, humans might not have the ability to reason far ahead into the future [Kahneman, 2003] or might exhibit *present bias* [O'Donoghue and Rabin, 1999], giving stronger weights on immediate costs and benefits rather than balancing them against those in the future.

We study this two-party sequential decision making problem under the formulation of Markov decision process (MDP). A standard MDP is characterized by the set of states, the set of actions, the state transition function, and the reward function. The solution of an MDP is a *policy* that specifies which action to take in each state that maximizes the total reward. Our setting deviates from the standard MDP in two perspectives. First, there are two parties, a principal and an agent, in the same decision making environment. The principal and the agent share the same information about the state, state transition, and action set. However, they have different reward functions. Moreover, while the agent can take actions in the environment, the principal can only update the environment to influence the agent's actions. Second, the agent exhibits decision-making biases in his solution to the MDP. Since the focus of this paper is in sequential decision making, we focus on the *time-related decision biases*, including myopic decision making, bounded rationality, and present bias.

We consider two natural sets of design spaces that the principal can choose from to update the environment. In the first design space, the principal can modify the agent's reward function in MDP, and the agent's policy is based on the modified reward function. This design space corresponds to the

[1] We use *she* to denote the principal and *he* to denote the agent.

scenario in which the principal can update the environment in a global manner (e.g., changing the badge design in social networking sites), and the agent will take actions in the updated environment. In the second design space, when the agent is choosing an action during decision time, the principal can offer additional incentives to nudge the agent to choose a different action. This design space corresponds to the scenario in which the principal can take interventions during the agent's decision time (e.g., offering a coupon when the user navigates to a certain page). In environment design with both design spaces, the goal of the principal is to maximize her own total rewards, depending on the principal's reward function and the agent's actions, subject to a budget constraint that the amount of environment updates is limited.

We formulate the principal's environment design problems as constrained optimization problems under both design spaces. We first show that the optimization problems are generally NP-hard to solve for both design spaces. We then propose relaxed formulations and corresponding algorithms for solving the problems. To evaluate the effectiveness of our proposed algorithms for environment design, we conduct simulations to understand the algorithm performance over a range of scenarios and parameters. Moreover, to examine whether we can indeed update the environment to influence the decisions of real-world human decision makers, we conduct a human-subject experiment with 300 workers from Amazon Mechanical Turk. Our results demonstrate the environment updates derived by our algorithms can effectively influence humans' decisions and lead to better total payoff.

1.1 Related Work

Our work is built on the formulation of Markov decision process (MDP) commonly seen in reinforcement learning. Instead of solving the agent's optimal policy, we consider a Stackelberg game formulation, in which the principal first chooses how to update the environment, and then the agent makes decisions in the updated environment. The closest works that consider this two-party setting in MDP include policy teaching [Zhang and Parkes, 2008; Zhang *et al.*, 2009; Zhang *et al.*, 2018] and poisoning attack for reinforcement learning [Rakhsha *et al.*, 2020; Zhang *et al.*, 2020]. Our work deviates from these works by incorporating human behavioral models in the framework. The human models considered in this work are empirically motivated from behavioral economics, such as *bounded rationality* [Kahneman, 2003] and *present bias* [O'Donoghue and Rabin, 1999].

Our work joins the recent research theme that incorporates human models in computational frameworks [Frazier *et al.*, 2014; Mansour *et al.*, 2015; Tang and Ho, 2019; Tang and Ho, 2021; Kleinberg and Oren, 2014; Masters *et al.*, 2021a; Masters *et al.*, 2021b]. There have been other lines of research that also includes humans in the loop of reinforcement learning frameworks, such as inverse reinforcement learning [Ng *et al.*, 2000; Evans *et al.*, 2016; Shah *et al.*, 2019; Hughes *et al.*, 2020; Zhi-Xuan *et al.*, 2020] that infers the reward functions in MDP through (potentially human) demonstrations. Our work differs in that our goal is to induce humans to perform desired behavior through updating the environment instead of improving learning algorithms.

2 Problem Setting

Decision-making environment. We formulate the sequential decision making environment as a finite-time horizon MDP with two sets of reward functions: $W = \langle S, A, P, R^a, R^p, T\rangle$, where S is the set of states, A is the set of agent actions, $P(s'|s,a)$ is the transition probability from state s to state s' after taking action a, T is the time horizon, $R^a(s,a)$ is the bounded reward obtained by the agent after he takes action a at state s, and $R^p(s,a)$ is the bounded reward obtained by the principal after the agent takes a at state s.

Agent decision-making policy. Since the agent could be biased and might not make time-consistent decisions, we represent the agent policy in a time-inconsistent manner: $\Pi : S \times T \rightarrow A$. In particular, for an agent policy $\pi \in \Pi$, $\pi(s,t)$ denotes the action the agent will take in state s at time t when following policy π. We formulate the agent as a planner $H : W \rightarrow \Pi$, with input being an environment $w \in W$ and output being a policy $\pi \in \Pi$ according to his decision-making model. The agent's goal is to maximize his *perceived* (potentially *biased*) rewards. To characterize the time-inconsistent behavior of the agent, we define the notion $d(t)$, the discounting factor that the agent perceives the payoff obtained t steps ahead. In the standard setting, $d(t)$ is often assumed to be in the form of γ^t with $\gamma \in (0,1]$ being the time-discounting factor. In this paper, we address different forms of $d(t)$ that captures different agent models, which will be discussed later.

With $d(t)$ defined, we now characterize the agent policy by defining a *perceived* Q-function[2] $Q^\pi(s,a,t,\hat{t})$, specifying the agent's perceived value at time t for him to take action a in state s at a future time $t+\hat{t}$ and follows policy π afterwards. This additional $\hat{t}$ parameter captures the agent's time-inconsistent belief: what the agent *thinks* he will do in a future time $t+\hat{t}$ while at time t might be different from what he will actually do at time $t+\hat{t}$. We also abuse the notation and let $\pi(s,t,\hat{t})$ denote the action the agent thinks what he would do in state s in a future time $t+\hat{t}$ while at time t. This perceived $Q^\pi(s,a,t,\hat{t})$ can be expressed as the sum of (1) the perceived reward for taking action a in a future time step $t+\hat{t}$ while at time t: $d(\hat{t})R^a(s,a)$ and (2) the expected future reward for following policy π after $t+\hat{t}$: $\mathbb{E}[\sum_{t'=t+\hat{t}+1}^{T} d(t'-t)R^a(s^\pi_{t'}, \pi(s^\pi_{t'}, t, t'-t))]]$, where $s^\pi_{t'}$ is the random variable denoting the state at t' if the agent follows π after $t+\hat{t}$. The expectation is over the randomness of the state transition.

Since the policy is only executed with $\hat{t}=0$ ($\hat{t}>0$ represents the agent's belief of what he would do $\hat{t}$ steps ahead), we let $Q^\pi(s,a,t) = Q^\pi(s,a,t,0)$ and $\pi(s,t) = \pi(s,t,0)$. The agent policy π^* can then be written as:

$$\pi^*(s,t) = \operatorname*{argmax}_a Q^{\pi^*}(s,a,t) \tag{1}$$

For a given environment, the agent policy can be solved by applying standard techniques, such as backward induction.

[2] This definition extends the standard Q-function to incorporate the agent's biased decision making.

Biased agent models. As discussed above, we use the notion $d(t)$, denoting how much the agent discounts the payoff t steps in the future to characterize the agent's behavior. This notion characterizes many common behavioral models, with some illustrative examples below:

- Standard model: In the literature, the agent is often assumed to have a consistent time-discounting factor $\gamma \in (0, 1]$ for discounting future payoff. Therefore, we can set $d(t) = \gamma^t$ to represent this standard assumption.
- Bounded rationality or short-sightedness: It considers the scenario in which the agent can only perform limited computation due to either time, cognitive, or information constraints. This can be approximated by considering that the agent only has information or only can reason about information within τ steps. We can formulate this by setting $d(t) = \gamma^t$ for all $0 \leq t \leq \tau$, and $d(t) = 0$ for all $\tau < t \leq T$. In the special case of *myopic agent*, who only cares about the immediate payoff and not the future payoffs, we can set $\tau = 0$.
- Present bias: When choosing between earning 10 dollars 100 days from now or 11 dollars 101 days from now, most people will choose the latter. However, when again being asked to choose between earning 10 dollars now or 11 dollars tomorrow, many people will change their decisions. This example illustrates the *present bias*, describing humans' inconsistency in discounting future payoffs. One common way to account for this behavior is through hyperbolic discounting factor: $d(t) = \frac{1}{1+kt}$ for $k > 0$.

Design space of the principal. Recall that the principal aims to update the environment to influence the agent's actions. We consider two natural sets of "updates" the principal can make to the environment:

- Reward function modification: The principal may pay costs to modify the agent's reward function to influence the agent's decisions. Formally, the principal can modify the agent's reward from $R^a(s, a)$ to $\bar{R}^a(s, a) = R^a(s, a) + c(s, a)$ for taking action a in state s by paying a cost equal to the absolute value of the modification $|c(s, a)|$. The agent will only observe the modified reward function and will make decisions based on $\bar{R}^a$. Note that this type of environment updates is performed *offline* in the sense that it updates the environment before the agent starts to make their decisions in the environment.
- Action nudge: We also consider another design space, in which the principal can offer a non-negative incentive $c(s, a, t) \geq 0$ to *nudge* the agent to take action a in state s at time t. The agent's reward in state s would then be $R(s, a) + c(s, a, t)$ if taking action a at time t while the future perceived rewards do not change. Different from the reward function modification, this nudge influences the agent's decisions during *decision time*.

The principal's goal is to maximize her total rewards derived from the agent's actions under the budget constraint that the total cost does not exceed budget B. Given the agent's policy π and the initial state distribution $p_0(s)$, let $p_t^\pi(s)$ be the state distribution at time t when the agent follows policy π, the principal's total expected reward can be written as[3]:

$$\sum_{t=0}^{T} \sum_{s \in S} p_t^\pi(s) R^p(s, \pi(s, t)) \tag{2}$$

3 Problem Formulations and Algorithms

Before we formulate the environment design problems, we first present an important, although perhaps not surprising, result that if the agent exhibit biases in decision making, being oblivious of the biases could lead to undesired outcome for the principal. The result showcases the importance of taking human behavior into account in environment design[4].

Lemma 1. *If the principal performs environment design by assuming the agent is a standard agent while the agent is boundedly rational, the ratio between the principal's reward after environment design compared with the principal's reward obtained in environment design with the correct agent model could be arbitrarily close to* 0.

3.1 Reward Function Modification

We first consider the environment design problem in which the principal can influence the agent's decisions through modifying the agent's reward functions $R^a(s, a)$. Let $c(s, a)$ be the modification the principal makes on $R^a(s, a)$, and $\bar{R}^a(s, a) = R^a(s, a) + c(s, a)$ is the reward function that the agent perceives and based on when making decisions. Let the updated MDP environment be $\bar{w}$, replacing the agent reward function as $\bar{R}^a$, and the agent policy on this environment be $\pi = h(\bar{w})$. The environment design problem for the principal is to choose the set of updates $\{c(s, a)\}$ to maximize her payoff subject to the budget constraint B. Again, let the initial state distribution be $p_0(s)$, and $p_t^\pi(s)$ be the state distribution at time t when the agent follows policy π, we can formulate the environment design problem as follows,

$$\begin{aligned} \max_c & \sum_{t=0}^{T} \sum_{s \in S} p_t^\pi(s) R^p(s, \pi(s, t)) \\ \text{s.t.} & \sum_{s \in S} \sum_{a \in A} |c(s, a)| \leq B \; ; \pi = h(\bar{w}) \end{aligned} \tag{3}$$

Note that this is a bi-level optimization problem, in which the principal is optimizing over the space of $\{c(s, a)\}$ while the agent is optimizing his policy in response to the principal's update in the form of $\pi = h(\bar{w})$. To solve the inner optimization problem (the agent's optimal policy), we can define an updated $\bar{Q}^\pi$ by replacing the reward R^a with $\bar{R}^a$ and solve the policy π using backward induction. We show that this bi-level optimization problem is generally NP-hard to solve.

Theorem 2. *It is NP-hard to solve the environment design problem with reward function modification as defined in* (3).

[3] We do not include the time-discounting factor for the principal's payoff to simplify the presentations. Our results and discussion can be easily extended to the setting with time-discounting factor.

[4] All proofs are included in the appendix of the full paper.

Relaxed formulation. To address this hardness result, we propose to use a soft-max stochastic policy ρ to relax the deterministic policy π. This relaxation makes the inner optimization differentiable, so first-order optimization methods might be applied. Instead of using $\pi(s,t)$ to denote the chosen action, we use $\rho(s,a,t)$ to represents the probability of choosing action a in state s at time t. Moreover, we again use $\bar{Q}^\rho$ to denote the perceived cumulative reward for policy ρ. The definition is similar to Q^π except that we need to incorporate the randomness of policy when evaluating the future reward. Moreover, we use a soft-max form to approximate the agent policy: $\rho(s,a,t) = \frac{e^{\beta \bar{Q}^\rho(s,a,t)}}{\sum_{a'} e^{\beta \bar{Q}^\rho(s,a',t)}}, \forall s,a,t.$

Below we formulate the relaxed environment design problem. We now use $p_t^\rho(s)$ to denote the state distribution at time t (with $p_0^\rho(s)$ defined as the initial state distribution $p_0(s)$ for notational simplicity) when the agent follows policy ρ. In addition, we explicitly layout the state distribution over time following policy ρ as a constraint in the third constraint of the optimization problem. Since the gradient of the optimization variables exists, we can approach this optimization through a gradient-based algorithm, as in Algorithm 1.

$$\begin{aligned}
\max_c \; & \sum_{t=0}^{T} \sum_{s\in S} \sum_{a\in A} p_t^\rho(s) R^p(s,a) \rho(s,a,t) \\
\text{s. t.} & \sum_{s\in S} \sum_{a\in A} |c(s,a)| \le B \\
& \rho(s,a,t) = \frac{e^{\beta \bar{Q}^\rho(s,a,t)}}{\sum_{a'} e^{\beta \bar{Q}^\rho(s,a',t)}}, \forall s,a,t \\
& p_{t+1}^\rho(s) = \sum_{s'\in S} \sum_{a\in A} p_t^\rho(s') P(s|s',a) \rho(s',a,t), \forall s,t \\
& \rho(s,a,t) \ge 0, \forall s,a,t
\end{aligned} \tag{4}$$

Algorithm 1 Gradient-based Algorithm for Solving (4)

```
Input: learning rate $\delta$, maximal iterations $N$
initialize $c, i = 0$
while $i < N$ do
    sample $\hat{s} \in S, \hat{a} \in A$
    update $\bar{R}^a(s,a), \bar{Q}(s,a,t), \rho(s,a,t), p_t^\rho(s), \forall s,a,t$
    calculate $\frac{\partial \rho(s,a,t)}{\partial c(\hat{s},\hat{a})}, \frac{\partial p_t^\rho(s)}{\partial c(\hat{s},\hat{a})}, \forall s,a,t$
    $c(\hat{s},\hat{a}) \leftarrow c(\hat{s},\hat{a}) + \delta \frac{\partial \sum p_t^\rho(s) R^p(s,a) \rho(s,a,t)}{\partial c(\hat{s},\hat{a})}$
    $i \leftarrow i + 1$
end while
return c
```

Discussion. When $\beta \to \infty$, $\rho(s,a,t)$ approximates to a delta function with the probability mass on the action with the highest $\bar{Q}$ value, which recovers the original problem. Moreover, recall that the Q function is defined with respect to the policy (when calculating the expected future rewards). We can show that this soft-max relaxation converges to the Q function of deterministic policy exponentially fast in β. In our simulations, we also empirically demonstrate that setting a small β is enough to approximate the optimal of the original problem in (3).

Lemma 3. *For any environment w, let π_w and ρ_w be the agent's deterministic and stochastic policies following our model. Let $Q^{\pi_w}(s,a,t)$ and $Q^{\rho_w}(s,a,t)$ be the corresponding Q-functions. For all (s,a,t), we have*

$$|Q^{\pi_w}(s,a,t) - Q^{\rho_w}(s,a,t)| \le \mathcal{O}(e^{-\beta C}),$$

where $C > 0$ is a constant and β is the parameter of ρ.

3.2 Action Nudge

We now formulate the environment design problem via action nudge. The principal can choose to pay $c(s,a,t) \ge 0$ to the agent if he takes action a in state s at time t. In this approach, the agent's perceived Q function does not change, but the agent's action will be influenced by this additional incentive, i.e., the agent will choose the action that maximizes $Q^\pi(s,a,t) + c(s,a,t)$ in state s at time t. Moreover, since the nudge is calculated offline but deployed online, the budget constraint is satisfied in expectation. Formally, the principal's environment design problem can be written as:

$$\begin{aligned}
\max_c \; & \sum_{t=0}^{T} \sum_{s\in S} p_t^\pi(s) R^p(s, \pi(s,t)) \\
s.t. & \sum_{t=0}^{T} \sum_{s\in S} c(s, \pi(s,t), t) p_t^\pi(s) \le B \\
& \pi(s,t) = \operatorname*{argmax}_a \{Q^\pi(s,a,t) + c(s,a,t)\}, \forall s,t
\end{aligned} \tag{5}$$

Solving this problem directly is again generally NP-hard due to the same bi-level optimization property and the deterministic policy structure. Below we utilize the problem structure and develop an alternative formulation.

Alternative formulation. Let π be the agent's policy in the original decision-making environment. The goal of action nudge is to make the agent change from action $a = \pi(s,t)$ to a new action a'. Assume the principal can break ties in any way she prefers when multiple actions lead to the same payoff[5], the cost the principal needs to pay to make the agent select action a' instead of a is $c(s,a',t) = Q(s,a,t) – Q(s,a',t)$. We can pre-calculate all the cost the principal needs to pay for action nudge $c(s,a,t) = Q(s,\pi(s,t),t) - Q(s,a,t), \forall s,a,t$.

With the above observations and the additional tie-breaking assumption, the environment design problem via action nudge is reduced to selecting which action the principal should nudge the agent to select for all (s,t). The nudged action a would generate a reward of $R^p(s,a)$ and incurs a cost $c(s,a,t)$. The goal is to maximize the total rewards such that the total cost is no larger than budget B in expectation. This problem reduces to a standard constrained MDP problem.

[5] While this assumption seems strong, it can be approximately satisfied by adding a arbitrarily small value to $c(s,a',t)$ to make the agent break ties to align with the principal's goal.

$$
\begin{aligned}
&\max_{\phi} \sum_{t=0}^{T}\sum_{s\in S}\sum_{a\in A} R^p(s,a)\phi(s,a,t)\\
&s.t. \sum_{t=0}^{T}\sum_{s\in S}\sum_{a\in A} c(s,a,t)\phi(s,a,t) \le B\\
&\quad \sum_{s'\in S}\sum_{a\in A} P(s|s',a)\phi(s',a,t) = \sum_{a\in A}\phi(s,a,t+1), \forall s,t\\
&\quad \sum_{a\in A}\phi(s,a,0) = p_0(s), \forall s\\
&\quad \phi(s,a,t)\ge 0, \forall s,a,t
\end{aligned}
\tag{6}
$$

In this optimization problem, $\phi(s,a,t)$ is the optimization variables, representing the joint probability at time t for the agent to be in state s and take action a. To translate $\phi(s,a,t)$ to the stochastic policy $\rho(s,a,t)$, we have $\rho(s,a,t) = \frac{\phi(s,a,t)}{\sum_{a'\in A}\phi(s,a',t)}$. The optimization problem is a linear program in $\phi(s,a,t)$. Therefore we can directly apply standard linear programming solvers to solve this optimization problem. When the agent is in state s at time t, this solution indicates that the principal should nudge and offers $c(s,a,t)$ if $\phi(s,a,t) > 0$. [6]

4 Experiments

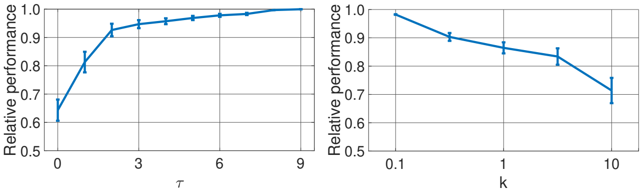
We conduct both simulated and real-human experiments to evaluate our proposed algorithms for environment design.

4.1 Simulations

In our simulations, we create a grid world of size 10×10. Each grid represents a state in the MDP. There are four actions representing the direction agent can move to: {up, down, left, right}. After each action, the agent moves to the nearby grid associated with the action with 70% chance and to a random nearby grid with 30% chance. The initial state is in the middle of the grid world. The time horizon T is set to be 20.

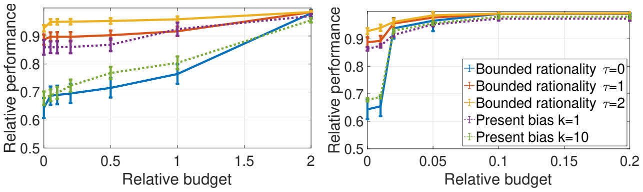
We initialize the principal's reward function values to be uniformly drawn from the range $[0, 0.5]$. We then randomly choose a 2×2 block as global optimal region and add 0.5 to the reward values within this block. Similarly, we randomly draw 1 to 3 local optimal regions (2×2 blocks) by setting their reward lower than global optimal but higher than its neighbors. We randomly generate 1,000 environments following the above procedure and report the average results. on these 1,000 environments.

Different agent behavioral models. We start with the setting that the agent's reward function is the same as the principal's, i.e., $R^p(s,a) = R^a(s,a)$ for all (s,a). In this setting, if the agent is behaving optimally, the principal does not need to update the environment. Therefore, we focus on examining how the agent's biased behavior impacts the total payoff and how effectively environment design can help.



(a) Bounded rationality. (b) Present biased.

Figure 1: The principal's payoff with biased decision-makers without environment design. Agents with higher τ or lower k are closer to being rational.



(a) Reward modification. (b) Action nudge.

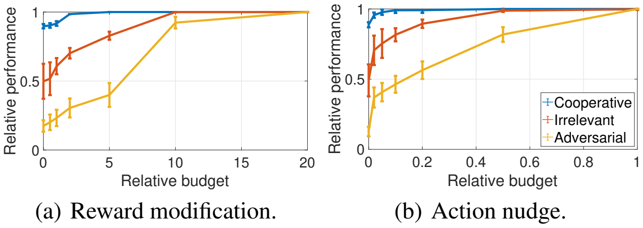
Figure 2: The principal's payoff with biased decision-makers after applying environment design. The y-axis is the relative performance compared with the optimal, and the x-axis is the amount of budget relative to the optimal performance.

We first examine the impact of biased agents without environment design. We consider agents with bounded rationality (or short-sightedness) and with present bias. Following the formulation in Section 2, we modify τ for boundedly-rational agents and k for present-bias agents. For boundedly-rational agents, we set $\gamma = 1$ and vary τ to be from 0 to 9. For present-bias agents, we vary k to be in $\{0.1, \sqrt{0.1}, 1, \sqrt{10}, 10\}$. The performance is measured in terms of the principal's objective. As shown in Figure 1, the principal's payoff, even when the reward function aligns with the agent's, could decrease significantly when the agent exhibits decision biases.

Next we examine the effect of environment design in improving the principal's payoff. We apply the algorithms in Section 3, with the soft-max parameter $\beta = 3$ (the choice of β is discussed in the appendix). We examine present-bias agents with $k \in \{1, 10\}$ and boundedly-rational agents with $\tau \in \{0, 1, 2\}$. We vary the budget for algorithms with both design spaces. As in Figure 2, our algorithms lead to effective environment design and improve with larger budget. While action nudge seems more cost efficient, the cost needs to be incurred for each agent. In reward modification, the environment may need only be updated once for multiple agents.

Mis-alignment of the principal's and the agent's objective. We now consider the case that the agent's reward function might not align with the principal's. We fix the principal's reward function as before and vary the agent's reward function. We consider the cases in which the agent's reward function is the inverse (adversarial), randomly drawn (irrelevant), and the same (cooperative) of the principal's reward function. The agent's bias model is set to be boundedly rational with $\tau = 1$ (the results are qualitatively similar for other agent models).

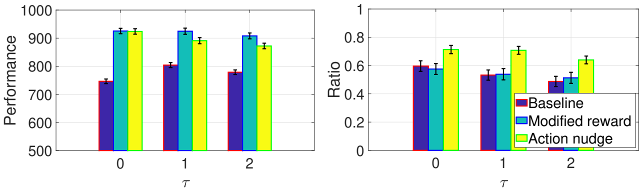
[6] There could be multiple actions that lead to $\phi(s,a,t) > 0$ for a given (s,t), leading to offering multiple nudges simultaneously. In Appendix C, we show that there exists a solution such that this does not happen frequently and discuss approaches to find this solution.



(a) Reward modification.

(b) Action nudge.

Figure 3: Misalignment of the principal's and the agent's the agent's reward function. The y-axis is the relative performance compared with the optimal (in terms of the principal's payoff), and the x-axis is the amount of budget relative to the optimal performance.



(a) Average principal's payoff.

(b) The ratio of human moves matched predictions.

Figure 4: The results from the human-subject experiment. The results are grouped by the vision length of the games, mapping to different values of τ in short-sighted (boundedly rational) agents. 4(a) shows the average principal's payoff with real human decision makers in treatments, and 4(b) shows the ratio of worker moves which are the same as short-sighted model predictions.

As shown in Figure 3, our algorithm can find the sets of environment updates to induce desired agent decisions, though it generally requires more budgets when the principal's reward function does not align with the agent's.

Additional simulations. Additional simulations are included in Appendix D. We show that setting a small β in Algorithm 1 suffices to approximate the true optimal of (3) and examine its runtime. This result complements Lemma 3 and demonstrates that we can approximate the overall performance of the optimal. In another simulation, we demonstrate how to combine off-the-shelf inverse reinforcement learning algorithms to deal with scenarios when the agent rewards and biases are unknown a priori.

4.2 Real-World Human-Subject Experiments

While our simulation results are promising, they are under the assumption that the agent makes decisions following the behavioral model. In this section, we examine whether our environment design algorithms are effective for real human decision makers whose behavior might deviate from the model. We have recruited 300 unique workers from Amazon Mechanical Turk. Each worker is paid \$0.50 and might earn additional bonuses. The average hourly rate is around \$11.50.

Task description. Each worker is asked to play six navigation games, with each represented by a grid world of size 10×10. The setup is similar to our simulations, except that we simplify the rewards to depend only on the state, i.e., $R^a(s,a) = R^p(s,a) = R(s)$, to reduce the cognitive burden for workers. Workers' bonuses depend on their total rewards. We also consider the setting in which the principal and the agent share the same reward function. To induce biased human behavior, a worker can only see the rewards of the nearby states (to simulate the short-sightedness). Out of six games, there are two games each for vision length of $1, 2, 3$, which we use short-sighted (boundedly rational) agent with $\tau = 0, 1, 2$ to model when solving the environment design problem. The detailed task setup is included in Appendix E.1.

Each worker is randomly assigned to one of the three treatments: {baseline, modified reward, action nudged}. The games are drawn from the same pool for each treatment. In baseline, workers play the drawn games without modifications. In modified reward, workers see the modified rewards generated by our algorithm. In action nudge, when a nudge happens, the workers see an additional messages indicating they might gain bonus for moving towards a certain direction. Since our goal is to observe whether environment design has impacts to real human decision-makers, we set the budget to be large enough such that the optimal decisions can be induced when the agent follows the behavioral model. We also report the true incurred cost in the experiment results.

Experiment results. As shown in Figure 4(a), workers under both environment design treatments generate more rewards for the principal, suggesting that our algorithms lead to effective environment designs even for real humans that do not always behave as the behavioral model. The actual costs incurred in "modified reward" and "action nudge" treatments are 73.7 and 50.3 points, while the average gain is 142.9 and 119.2 points. Moreover, since the principal and the agent share the same reward, the baseline treatment corresponds to the optimal design (do nothing) for the standard agent model. The performance improvement of our algorithms re-affirms the importance of incorporating realistic human models.

We also measure whether real humans behave as predicted by the behavioral model. As in Figure 4(b), worker behavior aligns with our behavioral models 53.8%, 54.2%, 68.7% of the time on average in each treatment. We also compare worker behavior with the standard model, with alignment at only 33.2%, 36.9%, 45.9% of the time. Interestingly, workers are more likely to behave as predicted in the "action nudge" treatment, likely because this treatment generates additional information that triggers workers to follow the nudged action.

5 Conclusion

We investigate environment design with biased decision makers. Our work sheds lights on many important applications, such as AI-assisted decision making. Future works include incorporating other bias models, including different environment design strategies, and addressing potential concerns when the objectives of the principal and the agent differ, such as in the adversarial setting. For example, can we design robust decision-making environments, e.g., imposing regulations/constraints on the environment updates to be allowed, to better safeguard human welfare. We hope this work can open more discussion in designing assistive AI technology and in incorporating behavioral models in computation.

Acknowledgements

This work is supported in part by the Office of Naval Research Grant N00014-20-1-2240.

References

[Abbeel and Ng, 2004] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

[Duan *et al.*, 2020] Xiaoni Duan, Chien-Ju Ho, and Ming Yin. Does exposure to diverse perspectives mitigate biases in crowdwork? an explorative study. In *AAAI Conference on Human Computation and Crowdsourcing*, 2020.

[Duan *et al.*, 2022] Xiaoni Duan, Chien-Ju Ho, and Ming Yin. The influences of task design on crowdsourced judgement: A case study of recidivism risk evaluation. In *The Web Conference (WWW)*, 2022.

[Evans and Goodman, 2015] Owain Evans and Noah D Goodman. Learning the preferences of bounded agents. In *NIPS Workshop on Bounded Optimality*, 2015.

[Evans *et al.*, 2016] Owain Evans, Andreas Stuhlmüller, and Noah Goodman. Learning the preferences of ignorant, inconsistent agents. In *AAAI Conference on Artificial Intelligence*, 2016.

[Frazier *et al.*, 2014] Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *ACM Conference on Economics and Computation*, 2014.

[Gottwald and Braun, 2019] Sebastian Gottwald and Daniel A Braun. Bounded rational decision-making from elementary computations that reduce uncertainty. *Entropy*, 2019.

[Hughes *et al.*, 2020] Dana Hughes, Akshat Agarwal, Yue Guo, and Katia Sycara. Inferring non-stationary human preferences for human-agent teams. In *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020.

[Kahneman, 2003] Daniel Kahneman. A perspective on judgment and choice: mapping bounded rationality. *American Psychologist*, 58(9):697, 2003.

[Kleinberg and Oren, 2014] Jon Kleinberg and Sigal Oren. Time-inconsistent planning: a computational problem in behavioral economics. In *ACM Conference on Economics and Computation*, 2014.

[Kleinberg *et al.*, 2017] Jon Kleinberg, Sigal Oren, and Manish Raghavan. Planning with multiple biases. In *ACM Conference on Economics and Computation*, 2017.

[Mansour *et al.*, 2015] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *ACM Conference on Economics and Computation*, 2015.

[Masters *et al.*, 2021a] Peta Masters, Michael Kirley, and Wally Smith. Extended goal recognition: a planning-based model for strategic deception. In *International Conference on Autonomous Agents and MultiAgent Systems*, 2021.

[Masters *et al.*, 2021b] Peta Masters, Wally Smith, and Michael Kirley. Extended goal recognition: Lessons from magic. *Frontiers in Artificial Intelligence*, 4, 2021.

[Ng *et al.*, 2000] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 2000.

[O'Donoghue and Rabin, 1999] Ted O'Donoghue and Matthew Rabin. Doing it now or later. *American Economic Review*, 89(1):103–124, 1999.

[Rakhsha *et al.*, 2020] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, 2020.

[Ramachandran and Amir, 2007] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2007.

[Schach *et al.*, 2018] Sonja Schach, Sebastian Gottwald, and Daniel A Braun. Quantifying motor task performance by bounded rational decision theory. *Frontiers in neuroscience*, 2018.

[Shah *et al.*, 2019] Rohin Shah, Noah Gundotra, Pieter Abbeel, and Anca Dragan. On the feasibility of learning, rather than assuming, human biases for reward inference. In *International Conference on Machine Learning*, 2019.

[Slivkins, 2019] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 2019.

[Song *et al.*, 2019] Zhao Song, Ron Parr, and Lawrence Carin. Revisiting the softmax bellman operator: New benefits and new perspective. In *International Conference on Machine Learning*, 2019.

[Tang and Ho, 2019] Wei Tang and Chien-Ju Ho. Bandit learning with biased human feedback. In *International Conference on Autonomous Agents and Multiagent Systems*, 2019.

[Tang and Ho, 2021] Wei Tang and Chien-Ju Ho. On the bayesian rational assumption in information design. In *AAAI Conference on Human Computation and Crowdsourcing*, 2021.

[Tang *et al.*, 2019] Wei Tang, Chien-Ju Ho, and Ming Yin. Leveraging peer communication to enhance crowdsourcing. In *The Web Conference (WWW)*, 2019.

[Zhang and Parkes, 2008] Haoqi Zhang and David C Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI Conference on Artificial Intelligence*, 2008.

[Zhang and Yu, 2013] Shunan Zhang and Angela J. Yu. Forgetful bayes and myopic planning: Human learning and decision-making in a bandit setting. In *Advances in Neural Information Processing Systems*, 2013.

[Zhang *et al.*, 2009] Haoqi Zhang, Yiling Chen, and David C Parkes. A general approach to environment design with one agent. In *International Joint Conference on Artificial Intelligence*, 2009.

[Zhang *et al.*, 2018] Haifeng Zhang, Jun Wang, Zhiming Zhou, Weinan Zhang, Ying Wen, Yong Yu, and Wenxin Li. Learning to design games: Strategic environments in reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2018.

[Zhang *et al.*, 2020] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, 2020.

[Zhi-Xuan *et al.*, 2020] Tan Zhi-Xuan, Jordyn Mann, Tom Silver, Josh Tenenbaum, and Vikash Mansinghka. Online bayesian goal inference for boundedly rational planning agents. In *Advances in Neural Information Processing Systems*, 2020.