# A General Approach to Environment Design with One Agent

**Haoqi Zhang, Yiling Chen, David Parkes**
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
{hq, yiling, parkes}@eecs.harvard.edu

## Abstract

The problem of environment design considers a setting in which an interested party aims to influence an agent's decisions by making limited changes to the agent's environment. Zhang and Parkes [2008] first introduced the environment design concept for a specific problem in the Markov Decision Process setting. In this paper, we present a general framework for the formulation and solution of environment design problems with one agent. We consider both the case in which the agent's local decision model is known and partially unknown to the interested party, and illustrate the framework and results on a linear programming setting. For the latter problem, we formulate an active, indirect elicitation method and provide conditions for convergence and logarithmic convergence. We relate to the problem of inverse optimization and also offer a game-theoretic interpretation of our methods.

## 1 Introduction

Many situations arise in which an interested party wants to influence an agent's behavior. A Web 2.0 site wants a user to contribute content. A teacher wants a student to follow an effective study behavior. A computer network operator wants to change the flow of traffic or contribution of resources. In the future, one can imagine the need to influence the behavior of robots as they interact with humans.

We are interested in problems in which the best way to modify the environment depends on information private to an agent, and in which the interested party can make limited changes to an agent's environment to encourage desired behaviors without directly mandating particular behaviors. A Web 2.0 site can tweak its user interface to ease the content uploading process or promote the contribution of one kind of content over another. A teacher can promote checking of answers by a student by providing gold stars or other positive reinforcements. A network operator can change latency or bandwidth availability on links or otherwise change the way that resources are allocated. One can place obstacles in an environment or change lighting conditions to perturb the plans generated by robots.

We view these problems as different examples of *environment design*. In environment design, one assumes an agent with a fixed way of making decisions and an interested party that can influence the agent's decisions via limited changes to the agent's environment. This is in contrast (and complementary) to much of artificial intelligence research, where the focus is on designing an agent that interacts with the environment. Environment design allows us to study settings where the interested party cannot design the agent but is still interested in its decisions, e.g. when the agent is human with inherent preferences and decision-making processes, or when the agent is a physical robot, whose hardware limits its physical and computational capabilities. By making limited changes to the environment, the interested party aims to align the agent's decisions under the modified environment with the decisions desired by the interested party.

In this paper, we consider an environment with one agent and one interested party. The agent makes decisions in the environment with respect to an agent function, which takes as input model parameters reflecting the agent's preferences and capabilities and the environment. The interested party seeks to affect the agent's decisions via changes to the environment, selected from a canonical set of changes that can depend on the agent's decisions and the environment. The goal of the interested party may be to elicit particular decisions from the agent, but can in general be expressed as a function of the decisions, model parameters, environment, and the environment change. The interested party knows the agent function and the environment, but in general does not know the model parameters. The interested party can observe the agent's decisions and interact multiple times with the agent, but this is the extent of the interested party's power.

Zhang and Parkes [2008] first introduced the environment design concept for a Markov Decision Process (MDP) domain, where they considered an interested party who can provide limited incentives to induce an agent to perform desired policies. In this work, we (1) provide a general framework for the formulation and solution of problems of environment design, and (2) illustrate the framework on a linear programming setting. We identify the different components of an environment design problem, and consider both the static case in which the agent's model parameters are known and the dynamic case in which they are partially unknown to the interested party. In the former case, we provide a general formulation of the environment design problem and relate to the problem of inverse optimization [Ahuja and Orlin, 2001]. In the latter case, we extend the active, indirect elicitation method proposed by Zhang and Parkes [2008] and provide conditions for convergence and logarithmic convergence. We conclude with a game-theoretic interpretation of our methods.[1]

### 1.1 Related work

Prior work on inverse optimization [Ahuja and Orlin, 2001] provides an answer for a special case of the static environment design problem where the goal of the interested party is to find minimal changes to a linear cost vector of an optimization problem so as to induce optimal decisions that coincide with specified target decisions. For the dynamic environment design problem, previous work provided an active indirect elicitation framework in the MDP domain, and established logarithmic convergence results for a simple goal function [Zhang *et al.*, 2009] and linear convergence results for a more complex goal function [Zhang and Parkes, 2008].

The environment design problem is similar in spirit to economic problems studied in principal-agent theory [Bolton and Dewatripont, 2005; Laffont and Martimort, 2001], which addresses the question of how a principal can provide incentives in the form of contracts to align the interest of an agent with that of the principal. But there are some fundamental differences between our problem and that of principal-agent theory. First, for the most part we assume that agent decisions are observable, and thus do not consider the "moral hazard" problem relating to hidden actions. Second, we consider repeated interactions with the agent, which allows us to overcome problems presented by the agent having private information about his or her preferences. Finally, we study problems in which the environment change is not limited to monetary incentives, and are generally willing to take a more indirect approach towards implementing desired outcomes.

In work on $k$-implementation, Monderer and Tennenholtz [2003] studied a problem in which an interested party assigns positive monetary rewards to influence the actions of agents in games. We can interpret $k$-implementation as a static multi-agent environment design problem in which the interested party has the power to assign unlimited rewards to states and can use commitment to implement desirable outcomes. The results in their paper are surprising in that monetary offers need not always materialize when agents follow desired behaviors, and may provide insights for studying other multi-agent environment design problems.

The problem of environment design is also related to mechanism design problems,where an interested party aims to design rules of a game to achieve desired outcomes [Jackson, 2003]. The heart of the mechanism design problem is in eliciting private information from agents so that the center can make correct system-wide decisions. This is in contrast to environment design, where private information is indirectly inferred from observing agents over repeated interactions and the agents ultimately take actions, not the center.

## 2 The Single-Agent Environment Design Problem

An environment design problem consists of the following components: an environment, an agent model, an environment change, an admissibility condition, an environment transition function, and a goal function. Below we provide definitions describing each of these components, and then present a static and dynamic formulation of the problem.

### 2.1 Problem setup

We consider an agent that acts in an *environment* $e \in \mathcal{E}$ based on its agent model $\mathcal{M} = \{\theta, f\}$, which consists of the *model parameters* $\theta \in \mathcal{I}$ and the *agent function* $f : \mathcal{I} \times \mathcal{E} \to 2^{\mathcal{X}}$. The model parameters represent the agent's preferences and capabilities, and the agent function takes the model parameters and environment as input and identifies (perhaps multiple, equivalent) decisions from the decision space $\mathcal{X}$. Note that while the agent may be indifferent among a set of decisions, the interested party may have different values for decisions in the set.

We make a couple of assumptions about the agent function $f$. First, we assume that the agent fully perceives aspects of the environment relevant to its decisions.[2] Second, we assume that $f$ is fixed and known to the interested party. Third, we assume the agent can compute $f$ on any input it encounters, such that any computational limitations of the agent is embedded within $f$. Lastly, we assume that the agent will deterministically pick a single decision $x \in f(\theta, e)$ when $f$ returns a non-singleton set of decisions, with this tie-breaking rule a priori unknown to the interested party.

We turn to consider the interested party's problem. The interested party knows the environment $e$, and can modify it via an *environment change* $\Delta \in \mathbf{\Delta}$. Given a set $X \in 2^{\mathcal{X}}$ of possible agent decisions, the admissible set *admissible*$_e(X) \subseteq \mathbf{\Delta}$ characterizes the space of allowable environment changes. Admissibility conditions model the interested party's design costs and constraints, both of which may depend on the agent's decisions. We assume the admissible set contains a null element $\Phi$ corresponding to no environment change.

The *environment transition function* $\mathcal{F} : \mathcal{E} \times \mathbf{\Delta} \to \mathcal{E}$ takes the current environment and an environment change and outputs the modified environment. We assume that the function is deterministic and known to the interested party. The agent then makes decisions in the modified environment, which enters as input to the agent function.

Finally, we define the goal of the interested party. The *goal function* $\mathcal{G} : \mathcal{X} \times \mathbf{\Delta} \times \mathcal{I} \times \mathcal{E} \to \Re$ takes the agent decision under the modified environment, the environment change, the agent's model parameters (a priori unknown in general), and the environment and outputs the value to the interested party. In addition to depending on the decision of the agent, the goal may depend on: the environment change because the interested party may care about the cost of the environment change; the model parameters because the designer may be altruistic, e.g., wish to maximize the value to the agent; and

[1] Most proofs are omitted in the interest of space and will be made available in a longer version of the paper.

[2] Alternatively, one can define $f$ based on the agent's perceptual inputs as opposed to the environment. For sake of exposition we do not explicitly model perception in this paper.

the environment because the interested party may value the agent's decisions differently in different environments. *The interested party aims to find an admissible $\Delta$ such that the agent's elicited behavior in the modified environment maximizes the goal function $\mathcal{G}$.*[3]

For the rest of this paper, we will only consider goal functions of the form $\mathcal{G} : \mathcal{X} \times \mathbf{\Delta} \rightarrow \Re$. A particular goal of interest is eliciting a desired target decision $x_T$, and this can be represented by a *single-configuration goal function*, where $\mathcal{G}(x, \Delta) > 0$ for target decision $x = x_T$ and is 0 otherwise. Here the value of the goal function can vary with $\Delta$, which allows the interested party to represent preferences over different environment changes leading to the same desired decision. A special case of the single-configuration goal function is the *indicator goal function*, for which $\mathcal{G}(x, \Delta) = 1$ for $x = x_T$ and is 0 otherwise. This can be used to represent situations where the interested party only cares about the elicited decisions and not the precise cost of the environment change.

The environment enters as input into the agent function and thus modifying the environment may influence the agent's decisions. Our base assumption is that the agent is myopic with respect to environment changes, e.g. the agent follows its agent function and does not reason about future changes to the environment when making current decisions. This assumption is restrictive in the dynamic formulation, where the agent's decisions influence future environment changes as the interested party is learning about the model parameters. In Section 4, we discuss conditions on the interested party's problem under which the assumption can be relaxed.

Before moving on, we consider as an example an interested party who wishes to design the user interface of a Web 2.0 site. The table below illustrates the various components of the problem under the environment design framework:

| | |
|---|---|
| Environment | a Web 2.0 site |
| Agent model parameters | preferences over site modules; time available to spend online |
| Agent function | perform actions on site to maximize utility |
| Environment change | adding, removing, and moving modules in the user interface |
| Admissibility condition | limit to changes within template; keep main components |
| Env. transition function | describes how the user interface changes |
| Goal function | values user being on the site and contributing content |

## 2.2 The Static Formulation

In the *static formulation* of the environment design problem, the agent's model parameters are assumed known to the interested party. Since we already assume that the interested party knows the agent function and the environment, we can think of the interested party's problem as one of *inverse optimization* [Ahuja and Orlin, 2001]. A key component of solving inverse problems is to characterize the space of inputs for which the agent function outputs a particular (desirable) set of decisions. We call this the *inverse feasible space*:

**Definition 1.** The model parameters $\theta$ and environment $e$ are *inverse feasible* with respect to an agent function $f$ and decisions $x$ if and only if $x \in f(\theta, e)$. Let $f^{-1}(x)$ denote the *inverse feasible space*, such that $(\theta, e) \in f^{-1}(x)$ if and only if $x \in f(\theta, e)$.

Given a characterization of the inverse feasible space, we need to ensure that the environment change $\Delta$ leads to a modified environment that together with the model parameters are inverse feasible for the desired decision. However, this may not be sufficient because in the case of multiple possible decisions in the range of the agent function, the agent may not select the one desired by the interested party. To be certain that the agent selects decisions desired by the interested party, our formulation assumes that the agent selects the worst possible decision for the interested party's goal function:

**Definition 2.** Given an environment $e$, the static environment design problem is an inverse optimization problem to find an environment change $\Delta$ that maximizes the interested party's goal function in the worst case:

$$\max_{\Delta, e'} \min_{x_T} \mathcal{G}(x_T, \Delta) \tag{1a}$$

$$\text{subject to:} \quad e' = \mathcal{F}(e, \Delta) \tag{1b}$$

$$(\theta, e') \in f^{-1}(x_T) \tag{1c}$$

$$\Delta \in admissible_e(f(\theta, e')) \tag{1d}$$

In the case that the agent function outputs singleton decision sets, the objective of the optimization simplifies to $\max_{\Delta, x_T, e'} \mathcal{G}(x_T, \Delta)$.

The constraints ensure that $e'$ is the modified environment (1b), that the model parameters and modified environment is inverse feasible for some decisions $x_T$ (1c), and that the environment change is admissible with respect to the set of possible agent decisions (1d).

Whether or not this formulation is interesting depends on its tractability, which depends on the particular model and the form of the functions involved. For example, if the objective and all constraints of the inverse problem are linear, we have a linear program that can be solved using standard techniques, assuming there is not an exponential number of constraints. We return to this in Section 3.1.

## 2.3 The Dynamic Formulation

In the more interesting case, the agent's model parameters will initially be, at least partially, unknown to the interested party. Since the agent function depends on both the environment and the model parameters, an interested party cannot effectively modulate the environment to induce desired behavior without some knowledge of the model parameters. To address this, we consider the active, indirect elicitation paradigm of Zhang and Parkes [2008]. The interested party

[3] The interested party may have other preferences, for example to induce desired decisions after few interactions. For simplicity, we will not model these preferences explicitly but instead introduce methods for quickly eliciting desired behaviors.

has repeated interactions with the agent. In each interaction, the interested party modifies the environment and observes the agent's decisions in response to the modified environment. Each time the interested party changes the environment, the agent's modified decisions form an observation that, given the agent function is known and fixed, provides some evidence about the unknown model parameters.

We first define the space of possible model parameters consistent with past observations of agent behavior. Let $\mathcal{H}$ denote the history of past elicitation rounds, such that $(x^o, e^o) \in \mathcal{H}$ denotes observed decisions $x^o$ in environment $e^o$ in round $o$. Notice that the agent's actual model parameters $\theta^*$ satisfies $(\theta^*, e^o) \in f^{-1}(x^o)$ for all observations $(x^o, e^o) \in \mathcal{H}$. From this space, the interested party need only consider candidate model parameters for which there exists environment changes that may lead to higher values for the goal function. If no such parameters exist, the interested party must have already found the environment change that maximizes its goal function *with respect to the actual unknown model parameters*.

We define the notion of a *consideration space*:

**Definition 3.** For an environment $e$, history $\mathcal{H}$, and value bound $\underline{\mathcal{G}}$, the consideration space $\mathcal{K}(e, \mathcal{H}, \underline{\mathcal{G}})$ over variables $\theta \in \mathcal{I}, \Delta \in \mathbf{\Delta}, x_T \in \mathcal{X}, e' \in \mathcal{E}$ is given by:

$$e' = \mathcal{F}(e, \Delta) \tag{2a}$$

$$(\theta, e') \in f^{-1}(x_T) \tag{2b}$$

$$\Delta \in admissible_e(f(\theta, e')) \tag{2c}$$

$$(\theta, e^o) \in f^{-1}(x^o) \qquad \forall (x^o, e^o) \in \mathcal{H} \tag{2d}$$

$$\underline{\mathcal{G}} \leq \min_{x \in f(\theta, e')} \mathcal{G}(x, \Delta) \tag{2e}$$

Consider the problem of finding an environment change that leads to agent decisions with value at least $\underline{\mathcal{G}}$ to the interested party. We pick candidate model parameters $\hat{\theta}$ and a corresponding environment change $\hat{\Delta}$ from the consideration space $\mathcal{K}(e, \mathcal{H}, \underline{\mathcal{G}})$. The agent responds to the modified environment $e'$ based on its actual model parameters $\theta^*$, and outputs decision $x'$. If the goal value of $x'$ under $\hat{\Delta}$ is at least $\underline{\mathcal{G}}$, we have solved the problem. Otherwise, it must be that the candidate model parameters are not the actual model parameters, and that the added inverse feasibility constraint will remove it from the consideration space. In this case we repeat the process of choosing candidate model parameters and environment changes until the problem is solved. This method is captured in the procedure ELICITATLEAST($\underline{\mathcal{G}}$).

Given this method to find environment changes leading to decisions that meet a value bound, we can find the environment change that maximizes the goal function with respect to the agent's actual model parameters via search. We assume that the goal function is bounded above and below by $\mathcal{G}_u$ and $\mathcal{G}_l$, respectively, and use a binary search procedure which calls ELICITATLEAST as a subroutine. The procedure takes as input an optimality gap $\kappa > 0$, and terminates once we have found an environment change that is provably within $\kappa$ of the best achievable value based on the actual agent parameters. The elicitation algorithm is given in Algorithm 1.

Let $\mathcal{G}^{\theta}_{max}$ denote the solution to the static environment design problem with respect to the actual model parameters $\theta$.

**Algorithm 1** Binary Search Elicitation Algorithm

```
procedure BINARYSEARCHELICITATION($\mathcal{G}_l, \mathcal{G}_u, \kappa$)
    $low \leftarrow \mathcal{G}_l + \kappa$, $high \leftarrow \mathcal{G}_u$, $\Delta_{best} = \Phi$, $\mathcal{H} \leftarrow \emptyset$
    while $low < high$ do
        $mid \leftarrow (low + high)/2$
        if ELICITATLEAST($mid$) = FAILURE then
            $high \leftarrow mid$
        else
            $(\Delta, x') \leftarrow$ ELICITATLEAST($mid$)
            $low \leftarrow \mathcal{G}(x', \Delta) + \kappa$; $\Delta_{best} \leftarrow \Delta$
    return $\Delta_{best}$
procedure ELICITATLEAST($\underline{\mathcal{G}}$)
    repeat
        Choose $\hat{\theta}$ and $\hat{\Delta}$ from $\mathcal{K}(e, \mathcal{H}, \underline{\mathcal{G}})$
        if no such values exist then return FAILURE
        else
            Make environment change $\hat{\Delta}$; $e' \leftarrow \mathcal{F}(e, \hat{\Delta})$
            Observe $x'$ in $e'$; add $(x', e')$ to $\mathcal{H}$
    until $\mathcal{G}(x', \hat{\Delta}) \geq \underline{\mathcal{G}}$
    return $(\hat{\Delta}, x')$
```

We have the following results about the elicitation algorithm:

**Theorem 1.** *The binary search elicitation algorithm converges to an environment change $\Delta$ that leads to an agent decision with goal value at least $\mathcal{G}^{\theta}_{max} - \kappa$ for some chosen $\kappa > 0$ in no more than $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil + |\mathcal{I}|$ rounds, for the case that the model parameter space $\mathcal{I}$ is finite.*

Theorem 1 follows from the fact that after each round, either the value bound updates or some candidate model parameters are falsified and removed by the added inverse feasibility constraint. $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil$ corresponds to the maximum number of possible value bound updates in the binary search.

In the case of an uncountably infinite model parameter space, we assume the model parameter space is contained within a closed, bounded set in $\Re^{|\theta|}$, where $|\theta|$ represents the dimension of the parameter space. The space need not be connected, and can consist of the union of disjoint nonempty sets. We define the notion of an $\epsilon$-robust output:

**Definition 4.** $f(\theta, e)$ is an $\epsilon$-robust output for model parameter $\theta$, environment $e$, and $\epsilon > 0$ if for all $x \in f(\theta, e)$ and $q \in \Re^{|\theta|}$ such that $|q_i| \leq \frac{\epsilon}{2}$ for all $i$, $x \in f(\theta + q, e)$ if $(\theta + q) \in \mathcal{I}$.

We can view the hypercube of length $\epsilon$ centered at $\theta$ as a measure of the robustness of the agent's decisions given small changes in the model parameters.[4] For any $\epsilon > 0$, let $\mathcal{G}^{\theta}_{\epsilon}$ denote the value of the solution to the static environment design problem with the additional restriction that $f(\theta, e')$ is an $\epsilon$-robust output for modified environment $e'$. If no solutions exist under the restriction, $\mathcal{G}^{\theta}_{\epsilon}$ denotes the value of the agent's decisions under no environment change. We have the following theorem:

[4] All mentions of hypercubes refer to an axis-aligned hypercube. The choice of hypercubes is somewhat arbitrary; we could have instead used balls. Computationally speaking, hypercubes are convenient because they can be defined using a set of linear constraints.

**Theorem 2.** *Consider the binary search elicitation algorithm with the restriction that $f(\theta, e')$ is an $\epsilon$-robust output for some $\epsilon > 0$ added to the consideration space. Algorithm 1 converges to an environment change $\Delta$ that leads to an agent decision with goal value at least $\mathcal{G}^{\theta}_{\epsilon} - \kappa$ for some chosen $\kappa > 0$ in no more than $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil + N$ rounds for the case that the model parameter space $\mathcal{I}$ is contained within a closed, bounded set in $\Re^{|\theta|}$, where $N$ is finite and represents the maximum number of axis-aligned hypercubes with side length $\epsilon$ that $\mathcal{I}$ can hold with the property that no hypercube contains the center of another hypercube.*

The intuition behind Theorem 2 is that the $\epsilon$-robust output condition ensures that a hypercube of points centered at the candidate model parameters $\hat{\theta}$ is eliminated from future consideration when $\hat{\theta}$ is falsified. Since the space of model parameters is bounded, by a pigeonhole argument, the entire space is covered after a finite number of iterations.

Theorems 1 and 2 guarantee convergence regardless of the choice for $\hat{\theta}$ and $\hat{\Delta}$ from within a consideration space, and complement heuristic algorithms that may lead to fast convergence even in the absence of useful, provable bounds.[5] Of course, we are also interested in providing general conditions for guaranteeing logarithmic convergence. Consider the following lemma:

**Lemma 1.** *Assume that $f(\hat{\theta}, e')$ is an $\epsilon$-robust output for model parameters $\hat{\theta}$ and environment $e'$, and that $\hat{\theta}$ is falsified by observed decisions $x' \notin f(\hat{\theta}, e')$. Let $\Theta$ represent the space of model parameters consistent with $\mathcal{K}(e, \mathcal{H}, \underline{\mathcal{G}})$ for some $\mathcal{H}$ containing $(x', e')$ and value bound $\underline{\mathcal{G}}$. If $\Theta$ is convex, there exists a separating hyperplane between a hypercube with side length $\epsilon$ centered at $\hat{\theta}$ and $\Theta$.*

The hypercube of points around $\hat{\theta}$ and $\Theta$ are disjoint, and the lemma follows from the separating hyperplane theorem. For $\hat{\theta}$ falsified by observed decisions $x'$ in environment $e'$, let $P(x', e')$ denote such a separating hyperplane. Let $\bar{P}(x', e')$ denote a hyperplane that results from relaxing $P(x', e')$ in the direction perpendicular to $P(x', e')$ until it is arbitrarily close to $\hat{\theta}$. Let $\bar{H}(x', e')$ denote the halfspace without $\hat{\theta}$ defined by $\bar{P}(x', e')$. We define the *relaxed consideration space* $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ based on Definition 3, with the modification to include $\theta \in \bar{H}(x^o, e^o)$ instead of $(\theta, e^o) \in f^{-1}(x^o)$ in constraint 2d. Base on this, we define a modified version of Algorithm 1, denoted Algorithm 1*, where: (i) line 13 of Algorithm 1 uses $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ instead of $\mathcal{K}(e, \mathcal{H}, \underline{\mathcal{G}})$, (ii) line 17 of Algorithm 1 only adds $(x', e')$ to $\mathcal{H}$ if $\mathcal{G}(x', \hat{\Delta}) < \underline{\mathcal{G}}$, (iii) ElicitAtLeast returns FAILURE if it has not returned after $1 + \lceil \log_b \lceil (\frac{R}{\epsilon})^{|\theta|} \rceil \rceil$ iterations, where $b = \frac{1}{1-(1/c)}$ and $c$ is the base of the natural logarithm, and (iv) $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ also requires $f(\theta, e')$ to be an $\epsilon$-robust output for some $\epsilon > 0$. We have the following theorem:

**Theorem 3.** *Assume that all points wthin a hypercube of side length $\epsilon$ centered at the actual model parameters $\theta^*$ are in the model parameter space $\mathcal{I}$, and that $R$ is the minimal side length of an axis-aligned hypercube in $\Re^{|\theta|}$ that contains $\mathcal{I}$. Furthermore, assume that a separation oracle is available to provide the separating hyperplane $P(x', e')$ based on observed decisions $x'$ in environment $e'$. Let $\Theta^t$ denote the space of model parameters consistent with $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ in round $t$. If $\Theta^t$ is convex for all $t$, then picking $\hat{\theta}$ as the centroid of $\Theta^t$ at each round $t$ of Algorithm $1^*$ leads to an agent decision with goal value at least $\mathcal{G}^{\theta}_{2\epsilon} - \kappa$ for some chosen $\kappa > 0$ in no more than $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil \, (1 + \lceil \log_b \lceil (\frac{R}{\epsilon})^{|\theta|} \rceil \rceil)$ rounds.*

*Proof.* (sketch) Consider two rounds $t$ and $t+1$ within the same call to ElicitAtLeast($\underline{\mathcal{G}}$). Let $V^t$ denote the volume of $\Theta^t$ at the start of round $t$. We need only consider cases when the candidate model parameters are falsified by an added halfspace $\bar{H}(x', e')$ because otherwise we have an environment change and decisions that meet the value threshold. Since we pick $\hat{\theta}$ as the centroid of $\Theta^t$ and that any halfspace containing the centroid of a convex set in $\Re^n$ contains $1/c$ of its volume [Grunbaum, 1960], adding $\bar{H}(x', e')$ ensures that $V^{t+1} \leq (1 - \frac{1}{c})V^t$. Furthermore, since $P(x', e')$ separates $\Theta^{t+1}$ and a hypercube with side length $\epsilon$ centered at $\hat{\theta}$, $\bar{P}(x', e')$ must separate $\hat{\theta}$ and a hypercube with side length $\epsilon$ centered at any $\theta \in \Theta^{t+1}$. It follows that if $\theta^* \in \Theta^t$, all points within a hypercube $C$ of side length $\epsilon$ centered at $\theta^*$ will not be eliminated by $\bar{H}(x', e')$.

Since $\mathcal{G}^{\theta}_{2\epsilon}$ is reachable by $\theta^*$ for $2\epsilon$-robust output, it is also reachable by all points in $C$ for $\epsilon$-robust output because the hypercube with side length $\epsilon$ centered at each point in $C$ is contained within the hypercube with side length $2\epsilon$ centered at $\theta^*$. In the case where $\underline{\mathcal{G}} \leq \mathcal{G}^{\theta}_{2\epsilon}$, this implies that $C \subseteq \Theta^t$ for all $t$ within the call to ElicitAtLeast($\underline{\mathcal{G}}$). Since picking $\hat{\theta}$ from $C$ leads to agent decisions with value at least $\underline{\mathcal{G}}$, we will find an environment change meeting the value threshold in some round $k$ in which $V^k \geq \epsilon^{|\theta|}$. In the case where $\underline{\mathcal{G}} > \mathcal{G}^{\theta}_{2\epsilon}$, we do not have to meet the value threshold because $\underline{\mathcal{G}}$ is greater than the value the theorem ensures. Since $V^t \leq R^{|\theta|}$ for all $t$ and $\bar{H}(x', e')$ cuts off a constant fraction of the volume at each round, ElicitAtLeast will correctly return after at most $1 + \lceil \log_b \lceil (\frac{R}{\epsilon})^{|\theta|} \rceil \rceil$ iterations. Since the binary search ensures that we achieve goal value at least $\mathcal{G}^{\theta}_{2\epsilon} - \kappa$ after at most $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil$ calls to ElicitAtLeast, multiplying gives the desired bound. □

Theorem 3 guarantees logarithmic convergence to decisions with value at least $G^{\theta}_{2\epsilon}$, which depending on $\epsilon$ may be less than the value $G^{\theta}_{\epsilon}$ guaranteed by Theorem 2 for the same problem. Notice if the relaxed consideration space is convex for all goal values, then $\Theta^t$ is convex for all rounds $t$. The theorem also requires a separation oracle to find the separating hyperplane with the desired properties (the hyperplane always exists, by Lemma 1). When the space of model parameters consistent with the relaxed consideration space is characterized by linear constraints, efficient separation oracles exist [Vanderbei, 2008]. We return to this in Section 3.2.

While choosing the centroid is computationally expensive, we can estimate it by sampling in polynomial time and still get the logarithmic convergence result above with arbitrarily high probability [Bertsimas and Vempala, 2004].

[5] For example, Zhang and Parkes [2008] demonstrated fast convergence in an MDP setting despite having only a linear bound.

## 3 Application: Linear Programming Model

In this section we focus on models that are computationally tractable yet expressive enough to model real-world decision problems. Linear programming (LP) provides one such model, which finds a diverse range of applications to decision making problems in networks, production and inventory management, finance, resource allocation, and medicine. In a linear program, an agent makes a set of decisions to minimize a linear cost vector subject to a set of linear constraints. We consider linear programs of the following form:

**Definition 5.** An agent with an LP-based agent function $f$ solves the following linear program (henceforth **LP**):

$$\min_{\mathbf{x}} \textstyle\sum_{j\in J}(c_j + e_j)x_j \tag{3a}$$

$$\textstyle\sum_{j\in J} a_{ij}x_j \geq b_i \qquad \forall i \in I \tag{3b}$$

$$l_j \leq x_j \leq u_j \qquad \forall j \in J \tag{3c}$$

Here $\mathbf{x}$ is an $|J|$ dimensional vector of decision variables. The cost vector **c+e** models the agent's dissatisfaction with performing various tasks, and reflects both the agent's internal costs $\mathbf{c}$ and external costs $\mathbf{e}$ from the environment. Conceptually, we can think of the internal cost vector as abstractly representing the agent's preferences over decisions, whereas the external cost vector represents quantities such as time, distance, or monetary cost with measurable values that the interested party may be able to affect. The model parameters $\theta = (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{l}, \mathbf{u})$ include the internal cost vector $\mathbf{c}$, right-hand side vector $\mathbf{b}$, constraint matrix $\mathbf{a}$, and upper and lower bound vectors $\mathbf{u}$ and $\mathbf{l}$. $\theta$ reflects the agent's preferences (e.g., via the cost vector) and capabilities (e.g., via constraint coefficients and right hand side values). Given $\theta$ and $\mathbf{e}$, the agent function identifies a set of values for the decision variables that minimizes the objective subject to the constraints.

We consider an interested party who is able to *influence the agent's cost vector* by modifying the external cost vector $\mathbf{e}$. We let $\Delta$ be a vector of external cost changes, such that $\mathcal{F}(\mathbf{e}, \Delta) = \mathbf{e}' = \mathbf{e} + \Delta$. In specifying admissibility conditions on $\Delta$, we wish to capture constraints an interested party may face in changing the external cost vector. We let $D^L_j$ and $D^U_j$ represent the bounds on per-dimension cost changes, and let $D_{min}$ and $D_{max}$ represent bounds on the total budget.[6] Consider the following definition:

**Definition 6.** Let $admissible_{LP}(X_T)$ denote the set of admissible external cost changes with respect to a set of target decisions $X_T$. $\Delta \in admissible_{LP}(X_T)$ if:

$$D^L_j \leq \Delta_j \leq D^U_j \qquad \forall j \in J \tag{4a}$$

$$D_{min} \leq \textstyle\sum_j \Delta_j x^j_T \leq D_{max} \qquad \forall x_T \in X_T \tag{4b}$$

Note that the admissibility conditions are defined with respect to induced agent decisions because the cost of the environment change may depend on the agent's decisions.

### 3.1 Solving the Static LP Formulation

We first consider the static case in which the interested party knows the model parameters and the external cost vector. The following linear constraints characterize the inverse feasible space of cost vectors for which a decision is optimal:

**Theorem 4.** *[Ahuja and Orlin, 2001] Consider decision $x_T$ that is feasible for* **LP**. *Let $f^{-1}_{\mathbf{LP}}(x_T)$ denote the space of parameter/environment pairs for which $x_T$ is optimal. For $\theta = (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{l}, \mathbf{u})$, $(\theta, \mathbf{e}) \in f^{-1}_{\mathbf{LP}}(x_T)$ if and only if:*

$$\textstyle\sum_{i\in B} a_{ij}\pi_i + \lambda_j = c_j + e_j \qquad \forall j \in L \tag{5a}$$

$$\textstyle\sum_{i\in B} a_{ij}\pi_i - \phi_j = c_j + e_j \qquad \forall j \in U \tag{5b}$$

$$\textstyle\sum_{i\in B} a_{ij}\pi_i = c_j + e_j \qquad \forall j \in F \tag{5c}$$

$$\pi_i \geq 0 \qquad \forall i \in I \tag{5d}$$

$$\lambda_j, \phi_j \geq 0 \qquad \forall j \in J \tag{5e}$$

*where $B = \{i \in I : \sum_{j\in J} a_{ij}x_j = b_i\}$, $L = \{j \in J : x^j_T = l_j\}$, $U = \{j \in J : x^j_T = u_j\}$, $F = \{j \in J : l_j < x^j_T < u_j\}$, and $\pi_i, \lambda_j, \phi_j$ represent the dual variables of* **LP**.

We restrict our attention to single configuration goal functions with desired target decision $x_T$. Since only $x_T$ provides positive value for the interested party and we aim to maximize the goal function in the worst case, we can restrict our attention to external cost changes $\Delta$ for which $x_T$ is uniquely optimal with respect to $\theta$ and $\mathbf{e} + \Delta$. Intuitively, a solution is unique for a linear program if the solution is robust to small changes to the 'slope' of the objective. More formally:

**Theorem 5.** *[Mangasarian, 1979] An optimal solution $x$ of* **LP** *is unique if and only if for each $\mathbf{q}$ in $\Re^{|J|}$ there exists an $\epsilon > 0$ such that $x$ remains a solution of the perturbed linear program with cost vector $\mathbf{c} + \mathbf{e} + \epsilon\mathbf{q}$.*

Theorem 5 is difficult to apply directly because the uniqueness condition is specified based on an infinite number of perturbed LPs. We show it is sufficient that the solution remains optimal for perturbations along each coordinate directions:

**Lemma 2.** *Let $\mathbf{v}^{\mathbf{j}}_\epsilon$ denote a $|J|$-dimensional vector with $\frac{\epsilon|J|}{2} + k$ in the $j$-th coordinate and 0 elsewhere for some $\epsilon > 0$ and arbitrarily small $k > 0$. If $(\theta, \mathbf{e}') \in f^{-1}_{\mathbf{LP}}(x_T)$ and $(\theta, \mathbf{e}' + \mathbf{v}^{\mathbf{j}}_\epsilon), (\theta, \mathbf{e}' - \mathbf{v}^{\mathbf{j}}_\epsilon) \in f^{-1}_{\mathbf{LP}}(x_T)$ for all $j \in J$, decision $x_T$ is uniquely optimal for* **LP** *with cost vector $\mathbf{c} + \mathbf{e}'$. Furthermore, $x_T$ is uniquely optimal for* **LP** *with any cost vectors within a hypercube with side length $\epsilon$ centered at $\mathbf{c} + \mathbf{e}'$.*

We formulate the static LP environment design problem:

**Theorem 6.** *For any single-configuration goal function $\mathcal{G}$ with desired decisions $x_T$ that can be expressed via a linear objective (with a polynomial number of corresponding linear constraints), the static environment design problem can be solved in polynomial time with respect to the size of* **LP** *using the following linear program for some small $\epsilon > 0$:*

$$\max_{\Delta} \mathcal{G}(x_T, \Delta) \tag{6a}$$

$$(\theta, \mathbf{e} + \Delta) \in f^{-1}_{\mathbf{LP}}(x_T) \tag{6b}$$

$$(\theta, \mathbf{e} + \Delta + \mathbf{v}^{\mathbf{j}}_\epsilon) \in f^{-1}_{\mathbf{LP}}(x_T) \qquad \forall j \in J \tag{6c}$$

$$(\theta, \mathbf{e} + \Delta - \mathbf{v}^{\mathbf{j}}_\epsilon) \in f^{-1}_{\mathbf{LP}}(x_T) \qquad \forall j \in J \tag{6d}$$

$$\Delta \in admissible_{LP}(\{x_T\}) \tag{6e}$$

[6] Our admissibility definition allows for situations where the interested party is not allowed to increase the cost (e.g., $D^U_j = 0$ for all $j$), or change the cost along a particular dimension (e.g. $D^L_k = D^U_k = 0$ for some $k$). This is often true in practice, where the interested party can only affect cost parameters along some but not all dimensions of the agent's decision.

Many useful single configuration goal functions such as minimizing the total cost change with respect to the $L_1$ or $L_\infty$ norms can be expressed as linear objectives and hence have tractable static environment design problems.

### 3.2 Solving the Dynamic LP Formulation

In the dynamic case, the agent's internal cost vector $\mathbf{c}$ is initially unknown to the interested party. We adopt Algorithm $1^*$ for this setting; here we (i) let ELICITATLEAST return FAILURE after $1 + \lceil \log_b \lceil (\frac{R}{\epsilon})^{|J|} \rceil \rceil$ iterations (since $|J|$ is the dimension of $\mathbf{c}$), (ii) add constraints 6c and 6d to $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ instead of requiring $\epsilon$-robust output, and (iii) for falsified cost vector $\hat{\mathbf{c}}$, find $P(x', e')$ based on $\Theta$ and the set $S_{\hat{\mathbf{c}}} = \{\hat{\mathbf{c}} + \sum_{j \in J} (\alpha_j^+ - \alpha_j^-) \mathbf{v}_\epsilon^\mathbf{j} | \sum_{j \in J} \alpha_j^+ + \alpha_j^- \leq 1, \alpha_j^-, \alpha_j^+ \geq 0\}$.

**Theorem 7.** *Assume that the actual internal cost vector* $\mathbf{c}^*$ *and points in* $S_{\mathbf{c}^*}$ *are in the internal cost vector space* $\mathcal{I}_\mathbf{c}$*, and that* $R$ *is the minimal side length of an axis-aligned hypercube in* $\Re^{|J|}$ *that contains* $\mathcal{I}_\mathbf{c}$*. Let* $\mathcal{G}_\epsilon^{LP}$ *denote the value of the solution to the LP in Theorem 6 with respect to* $c^*$ *and some* $\epsilon > 0$ *if a solution exists, and the goal value with no environment change otherwise. Assume the interested party has a linear single configuration goal function, and consider Algorithm* $1^*$ *for the LP setting. Let* $\Theta^t$ *denote the space of internal cost vectors consistent with* $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ *in round* $t$*. Then picking* $\hat{\mathbf{c}}$ *as the centroid of* $\Theta^t$ *at each round* $t$ *leads to an agent decision with goal value at least* $\mathcal{G}_{2\epsilon}^{LP} - \kappa$ *for some chosen* $\kappa > 0$ *in no more than* $\lceil \log_2 \lceil \frac{\mathcal{G}_u - \mathcal{G}_l}{\kappa} \rceil \rceil \, (1 + \lceil \log_b \lceil (\frac{R}{\epsilon})^{|J|} \rceil \rceil)$ *rounds.*

Theorem 7 follows from applying Theorem 3 to this setting. Notice that $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ in the LP setting for single configuration goal functions is a convex polytope characterized by linear constraints. This implies that $\Theta^t$ is convex and allows one to find a separating hyperplane between this set and $S_{\hat{\mathbf{c}}}$ by solving a linear program (e.g., see Theorem 10.4 of [Vanderbei, 2008]). Since picking $\hat{\theta}$ and $\hat{\Delta}$ from $\bar{\mathcal{K}}(e, \mathcal{H}, \underline{\mathcal{G}})$ need only require solving an LP, Algorithm $1^*$ can be computed in polynomial time.

## 4 A Game-theoretic Interpretation

Our formulations and results can be interpreted as equilibria of simple games. We can view the static formulation as an extensive form game in which the interested party first chooses an environment change $\Delta$ and the agent then makes decisions in the modified environment. From this perspective, the agent function is a best response correspondence to the environment and model parameters. We can view the interested party's environment change based on Definition 2 and the agent's response as strategies that form a subgame perfect Nash equilibrium of this game. Similarly, we can view the dynamic formulation as a repeated game whose stage extensive-form game is as in the static case. Here a myopic agent will still best respond with its agent function. For an interested party following the elicitation strategy, the agent's best response leads to maximizing the interested party's goal.

If the agent were forward looking, the game-theoretic analysis becomes more complicated because the agent may prefer certain environments than others and seek to influence the interested party's choice of environment changes. In the special case that the interested party has an indicator goal function and any environment change is preferred to the current environment, the agent may still try to influence the interested party's choices but will want to make sure that the interested party does not give up on eliciting decisions so that environment changes will continue to be provided. It can be shown that the agent's best response leads to the desired decisions if the agent is sufficiently patient (e.g., see [Zhang *et al.*, 2009]).

## 5 Conclusion

We formulated a general problem of environment design for a single agent and provided an algorithm with logarithmic convergence results for the dynamic case in which an agent's model parameters are initially unknown to the interested party. We developed polynomial time solutions for an application to environment design with linear-programming based agent functions. Future work should continue to explore game-theoretic aspects of environment design, extend to richer goal functions, consider partial observations, allow for multiple agents, and provide other specific applications of modifying agent environments to achieve design goals.

## Acknowledgements

The first author acknowledges support from an NDSEG fellowship. The authors gratefully acknowledge useful feedback from anonymous reviewers on an earlier version of the paper.

## References

[Ahuja and Orlin, 2001] Ravindra K. Ahuja and James B. Orlin. Inverse optimization. *Operations Research*, 49:771–783, 2001.

[Bertsimas and Vempala, 2004] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[Bolton and Dewatripont, 2005] Patrick Bolton and Mathias Dewatripont. *Contract Theory*. MIT Press, 2005.

[Grunbaum, 1960] Branko Grunbaum. Partitions of mass-distributions and of convex bodies by hyperplanes. *Pacific Journal of Mathematics*, 10(4):1257–1261, 1960.

[Jackson, 2003] Matthew O. Jackson. Mechanism theory. In Ulrich Derigs, editor, *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2003.

[Laffont and Martimort, 2001] Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, 2001.

[Mangasarian, 1979] O. L. Mangasarian. Uniqueness of solution in linear programming. *Linear algebra and its applications*, 25:151–162, 1979.

[Monderer and Tennenholtz, 2003] Dov Monderer and Moshe Tennenholtz. k-implementation. In *EC '03: Proc. 4th ACM conference on Electronic Commerce*, pages 19–28, 2003.

[Vanderbei, 2008] Robert J. Vanderbei. *Linear programming : foundations and extensions*. Springer, 3rd edition, 2008.

[Zhang and Parkes, 2008] Haoqi Zhang and David Parkes. Value-based policy teaching with active indirect elicitation. In *Proc. 23rd National Conference on Artificial Intelligence*, 2008.

[Zhang *et al.*, 2009] Haoqi Zhang, David Parkes, and Yiling Chen. Policy teaching through reward function learning. In *Proc. 10th ACM Conference on Electronic Commerce (EC'09)*, 2009.