Safe Policy Improvement by Minimizing Robust Baseline Regret

Marek Petrik
University of New Hampshire
mpetrik@cs.unh.edu

Mohammad Ghavamzadeh
Adobe Research & INRIA Lille
ghavamza@adobe.com

Yinlam Chow
Stanford University
ychow@stanford.edu

Abstract

An important problem in sequential decision-making under uncertainty is to use limited data to compute a *safe* policy, which is guaranteed to outperform a given baseline strategy. In this paper, we develop and analyze a new *model-based* approach that computes a safe policy, given an inaccurate model of the system's dynamics and guarantees on the accuracy of this model. The new robust method uses this model to directly minimize the (negative) regret w.r.t. the baseline policy. Contrary to existing approaches, minimizing the regret allows one to improve the baseline policy in states with accurate dynamics and to seamlessly fall back to the baseline policy, otherwise. We show that our formulation is NP-hard and propose a simple approximate algorithm. Our empirical results on several domains further show that even the simple approximate algorithm can outperform standard approaches.

1 Introduction

Many problems in science and engineering can be formulated as a sequential decision-making problem under uncertainty. A common scenario in such problems that occurs in many different fields, such as online marketing, inventory control, health informatics, and computational finance, is to find a good or an optimal strategy/policy, given a batch of data generated by the current strategy of the company (hospital, investor). Although there are many techniques to find a good policy given a batch of data, only a few of them guarantee that the obtained policy will perform well, when it is deployed. Since deploying an untested policy can be risky for the business, the product (hospital, investment) manager does not usually allow it to happen, unless we provide her/him with some performance guarantees of the obtained strategy, in comparison to the baseline policy (for example the policy that is currently in use).

In this paper, we focus on the *model-based* approach to this fundamental problem in the context of *infinite-horizon* discounted Markov decision processes (MDPs). In this approach, we use the batch of data and build a *model* or a *simulator* that approximates the true behavior of the dynamical system, together with an *error function* that captures the accuracy of the model at each state of the system. Our goal is to compute a *safe* policy, i.e., a policy that is guaranteed to perform at least as well as the baseline strategy, using the simulator and error function. Most of the work on this topic has been in the *model-free* setting, where *safe* policies are computed directly from the batch of data, without building an explicit model of the system [Thomas *et al.*, 2015b,a]. Another class of *model-free* algorithms are those that use a batch of data generated by the current policy and return a policy that is guaranteed to perform better. They optimize for the policy by repeating this process until convergence [Kakade and Langford, 2002; Pirotta *et al.*, 2013].

A major limitation of the existing methods for computing safe policies is that they either adopt a newly learned policy with provable improvements or do not make any improvement at all by returning the baseline policy. These approaches may be quite limiting when model uncertainties are not uniform

across the state space. In such cases, it is desirable to guarantee an improvement over the baseline policy by combining it with a learned policy on a state-by-state basis. In other words, we want to use the learned policy at the states in which either the improvement is significant or the model uncertainty (error function) is small, and to use the baseline policy everywhere else. However, computing a learned policy that can be effectively combined with a baseline policy is non-trivial due to the complex effects of policy changes in an MDP. Our key insight is that this goal can be achieved by minimizing the (negative) *robust regret* w.r.t. the baseline policy. This unifies the sources of uncertainties in the learned and baseline policies and allows a more systematic performance comparison. Note that our approach differs significantly from the standard one, which compares a pessimistic performance estimate of the learned policy with an optimistic estimate of the baseline strategy. That may result in rejecting a learned policy with a performance (slightly) better than the baseline, simply due to the discrepancy between the pessimistic and optimistic evaluations.

The *model-based* approach of this paper builds on *robust* Markov decision processes [Iyengar, 2005; Wiesemann *et al.*, 2013; Ahmed and Varakantham, 2013]. The main difference is the availability of the baseline policy that creates unique challenges for sequential optimization. To the best of our knowledge, such challenges have not yet been fully investigated in the literature. A possible solution is to solve the robust formulation of the problem and then accept the resulted policy only if its conservative performance estimate is better than the baseline. While similar idea has been investigated in the *model-free* setting (e.g., [Thomas *et al.*, 2015a]), we show in this paper that it can be overly conservative.

As the main contribution of the paper, we propose and analyze a new robust optimization formulation that captures the above intuition of minimizing robust regret w.r.t. the baseline policy. After a preliminary discussion in Section 2, we formally describe our model and analyze its main properties in Section 3. We show that in solving this optimization problem, we may have to go beyond the standard space of deterministic policies and search in the space of randomized policies; we derive a bound on the performance loss of its solutions; and we prove that solving this problem is NP-hard. We also propose a simple and practical approximate algorithm. Then, in Section 4, we show that the standard model-based approach is really a tractable approximation of robust baseline regret minimization. Finally, our experimental results in Section 5 indicate that even the simple approximate algorithm significantly outperforms the standard model-based approach when the model is uncertain.

2 Preliminaries

We consider problems in which the agent's interaction with the environment is modeled as an *infinite-horizon γ-discounted MDP*. A γ-discounted MDP is a tuple $\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, r, P, p_0, \gamma \rangle$, where $\mathcal{X}$ and $\mathcal{A}$ are the state and action spaces, $r(x, a) \in [-R_{\max}, R_{\max}]$ is the bounded reward function, $P(\cdot|x, a)$ is the transition probability function, $p_0(\cdot)$ is the initial state distribution, and $\gamma \in (0, 1]$ is a discount factor. We use $\Pi_R = \{\pi : \mathcal{X} \to \Delta^{\mathcal{A}}\}$ and $\Pi_D = \{\pi : \mathcal{X} \to \mathcal{A}\}$ to denote the sets of *randomized* and *deterministic* stationary Markovian policies, respectively, where $\Delta^{\mathcal{A}}$ is the set of probability distributions over the action space $\mathcal{A}$.

Throughout the paper, we assume that the true reward r of the MDP is known, but the true transition probability is not given. The generalization to include reward estimation is straightforward and is omitted for the sake of brevity. We use historical data to build a MDP *model* with the transition probability denoted by $\widehat{P}$. Due to limited number of samples and other modeling issues, it is unlikely that $\widehat{P}$ matches the true transition probability of the system $P^{\star}$. We also require that the estimated model $\widehat{P}$ deviates from the true transition probability $P^{\star}$ as stated in the following assumption:

Assumption 1. *For each $(x, a) \in \mathcal{X} \times \mathcal{A}$, the error function $e(x, a)$ bounds the ℓ_1 difference between the estimated transition probability and true transition probability, i.e.,*

$$\|P^{\star}(\cdot|x, a) - \widehat{P}(\cdot|x, a)\|_1 \leq e(x, a). \tag{1}$$

The error function e can be derived either directly from samples using high probability concentration bounds, as we briefly outline in Appendix A, or based on specific domain properties.

To model the uncertainty in the transition probability, we adopt the notion of robust MDP (RMDP) [Iyengar, 2005; Nilim and El Ghaoui, 2005; Wiesemann *et al.*, 2013], i.e., an extension of

MDP in which nature adversarially chooses the transitions from a given *uncertainty set*

$$\Xi(\widehat{P}, e) = \left\{ \xi : \mathcal{X} \times \mathcal{A} \to \Delta^{\mathcal{X}} : \|\xi(\cdot|x,a) - \widehat{P}(\cdot|x,a)\|_1 \le e(x,a), \ \forall x, a \in \mathcal{X} \times \mathcal{A} \right\}.$$

From Assumption 1, we notice that the true transition probability is in the set of uncertain transition probabilities, i.e., $P^\star \in \Xi(\widehat{P}, e)$. The above ℓ_1 constraint is common in the RMDP literature (e.g., [Iyengar, 2005; Wiesemann *et al.*, 2013; Petrik and Subramanian, 2014]). The uncertainty set Ξ in RMDP is (x, a)-rectangular and randomized [Le Tallec, 2007; Wiesemann *et al.*, 2013]. One of the motivations for considering (x, a)-rectangular sets in RMDP is that they lead to tractable solutions in the conventional reward maximization setting. However, in the robust regret minimization problem that we propose in this paper, even if we assume that the uncertainty set is (x, a)-rectangular, it does not guarantee tractability of the solution. While it is of great interest to investigate the structure of uncertainty sets that lead to tractable algorithms in robust regret minimization, it is beyond the main scope of this paper and we leave it as future work.

For each policy $\pi \in \Pi_R$ and nature's choice $\xi \in \Xi$, the discounted *return* is defined as

$$\rho(\pi, \xi) = \lim_{T \to \infty} \mathbb{E}_\xi \left[\sum_{t=0}^{T-1} \gamma^t r(X_t, A_t) \mid X_0 \sim p_0, \ A_t \sim \pi(X_t) \right] = p_0^\top v_\pi^\xi,$$

where X_t and A_t are the state and action random variables at time t, and v_π^ξ is the corresponding *value function*. An *optimal policy* for a given ξ is defined as $\pi_\xi^\star \in \arg\max_{\pi \in \Pi_R} \rho(\pi, \xi)$. Similarly, under the true transition probability $P^\star$, the *true return* of a policy π and a *truly optimal policy* are defined as $\rho(\pi, P^\star)$ and $\pi^\star \in \arg\max_{\pi \in \Pi_R} \rho(\pi, P^\star)$, respectively. Although we define the optimal policy using $\arg\max_{\pi \in \Pi_R}$, it is known that every reward maximization problem in MDPs has at least one optimal policy in Π_D.

Finally, given a deterministic *baseline* policy π_B, we call a policy π *safe*, if its "true" performance is guaranteed to be no worse than that of the baseline policy, i.e., $\rho(\pi, P^\star) \ge \rho(\pi_B, P^\star)$.

3 Robust Policy Improvement Model

In this section, we introduce and analyze an optimization procedure that robustly improves over a given baseline policy π_B. As described above, the main idea is to find a policy that is guaranteed to be an improvement for any realization of the uncertain model parameters. The following definition formalizes this intuition.

Definition 2 (The Robust Policy Improvement Problem). *Given a model uncertainty set $\Xi(\widehat{P}, e)$ and a baseline policy π_B, find a maximal $\zeta \ge 0$ such that there exists a policy $\pi \in \Pi_R$ for which $\rho(\pi, \xi) \ge \rho(\pi_B, \xi) + \zeta$, for every $\xi \in \Xi(\widehat{P}, e)$.*[1]

The problem posed in Definition 2 readily translates to the following optimization problem:

$$\pi_S \in \arg\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right). \tag{2}$$

Note that since the baseline policy π_B achieves value 0 in (2), ζ in Definition 2 is always non-negative. Therefore, any solution π_S of (2) is *safe*, because under the true transition probability $P^\star \in \Xi(\widehat{P}, e)$, we have the guarantee that

$$\rho(\pi, P^\star) - \rho(\pi_B, P^\star) \ge \min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right) \ge 0.$$

It is important to highlight how Definition 2 differs from the standard approach (e.g., [Thomas *et al.*, 2015a]) on determining whether a policy π is an improvement over the baseline policy π_B. The standard approach considers a statistical error bound that translates to the test: $\min_{\xi \in \Xi} \rho(\pi, \xi) \ge \max_{\xi \in \Xi} \rho(\pi_B, \xi)$. The uncertainty parameters ξ on both sides of (2) are not necessarily the same. Therefore, any optimization procedure derived based on this test is more conservative than the problem in (2). Indeed when the error function in Ξ is large, even the baseline policy ($\pi = \pi_B$)

---

[1] From now on, for brevity, we omit the parameters $\widehat{P}$ and e, and use Ξ to denote the model uncertainty set.
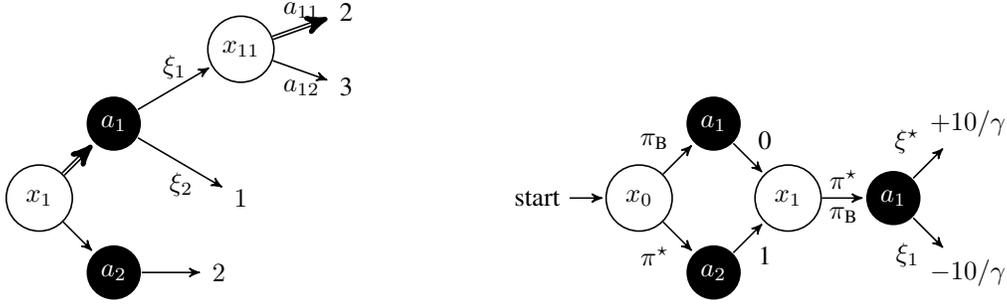
Figure 1: *(left)* A robust/uncertain MDP used in Example 4 that illustrates the sub-optimality of deterministic policies in solving the optimization problem (2). *(right)* A Markov decision process with significant uncertainty in the baseline policy.

may not pass this test. In Section 5.1, we show the conditions under which this approach fails. Our approach also differs from other related work in that we consider regret with respect to the baseline policy, and not the optimal policy, as considered in [Xu and Mannor, 2009].

In the remainder of this section, we highlight some major properties of the optimization problem (2). Specifically, we show that its solution policy may be purely randomized, we compute a bound on the performance loss of its solution policy w.r.t. $\pi^\star$, and we finally prove that it is a NP-hard problem.

3.1 Policy Class

The following theorem shows that we should search for the solutions of the optimization problem (2) in the space of randomized policies Π_R.

Theorem 3. *The optimal solution to the optimization problem* (2) *may not be attained by a deterministic policy. Moreover, the loss due to considering deterministic policies cannot be bounded, i.e., there exists no constant $c \in \mathbb{R}$ such that*

$$\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right) \leq c \cdot \max_{\pi \in \Pi_D} \min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right).$$

Proof. The proof follows directly from Example 4. The optimal policy in this example is randomized and achieves a guaranteed improvement $\zeta = 1/2$. There is no deterministic policy that guarantees a positive improvement over the baseline policy, which proves the second part of the theorem. □

Example 4. *Consider the robust/uncertain MDP on the left panel of Figure 1 with states $\{x_1, x_{11}\} \subset \mathcal{X}$, actions $\mathcal{A} = \{a_1, a_2, a_{11}, a_{12}\}$, and discount factor $\gamma = 1$. Actions a_1 and a_2 are shown as solid black nodes. A number with no state represents a terminal state with the corresponding reward. The robust outcomes $\{\xi_1, \xi_2\}$ correspond to the uncertainty set of transition probabilities Ξ. The baseline policy π_B is deterministic and is denoted by double edges. It can be readily seen from the monotonicity of the Bellman operator that any improved policy π will satisfy $\pi(a_{12}|x_{11}) = 1$. Therefore, we will only focus on the policy at state x_1. The robust improvement as a function of $\pi(\cdot|x_1)$ and the uncertainties $\{\xi_1, \xi_2\}$ is given as follows:*

$$\min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right) = \min_{\xi \in \Xi} \left(\begin{bmatrix} \pi \setminus \xi & \xi_1 & \xi_2 \\ \hline a_1 & 3 & 1 \\ a_2 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \pi \setminus \xi & \xi_1 & \xi_2 \\ \hline a_1 & 2 & 1 \end{bmatrix} \right) = 0.$$

This shows that no deterministic policy can achieve a positive improvement in this problem. However, a randomized policy $\pi(a_1|x_1) = \pi(a_2|x_1) = 1/2$ returns the maximum improvement $\zeta = 1/2$.

Randomized policies can do better than their deterministic counterparts, because they allow for hedging among various realizations of the MDP parameters. Example 4 shows a problem such that there exists a realization of the parameters with improvement over the baseline when any deterministic policy is executed. However in this example, there is no single realization of parameters that provides an improvement for all the deterministic policies *simultaneously*. Therefore, randomizing the policy guarantees an improvement independent of the parameters' choice.

3.2 Performance Bound

Generally, one cannot compute the truly optimal policy $\pi^\star$ using an imprecise model. Nevertheless, it is still crucial to understand how errors in the model translates to a performance loss w.r.t. an optimal policy. The following theorem (proved in Appendix C) provides a bound on the performance loss of any solution π_S to the optimization problem (2).

Theorem 5. *A solution π_S to the optimization problem* (2) *is safe and its performance loss is bounded by the following inequality:*

$$\Phi(\pi_S) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_S, P^\star) \leq \min\left\{\frac{2\gamma R_{\max}}{(1-\gamma)^2}\left(\|e_{\pi^\star}\|_{1, u^\star_{\pi^\star}} + \|e_{\pi_B}\|_{1, u^\star_{\pi_B}}\right), \Phi(\pi_B)\right\},$$

where $u^\star_{\pi^\star}$ and $u^\star_{\pi_B}$ are the state occupancy distributions of the optimal and baseline policies in the true MDP $P^\star$. Furthermore, the above bound is tight.

3.3 Computational Complexity

In this section, we analyze the computational complexity of solving the optimization problem (2) and prove that the problem is NP-hard. In particular, we proceed by showing that the following sub-problem of (2):

$$\arg\min_{\xi \in \Xi}\left(\rho(\pi, \xi) - \rho(\pi_B, \xi)\right), \tag{3}$$

for a fixed $\pi \in \Pi_R$, is NP-hard. The optimization problem (3) can be interpreted as computing a policy that simultaneously minimizes the returns of two MDPs, whose transitions induced by policies π and π_B. The proof of Theorem 6 is given in Appendix D.

Theorem 6. *Both optimization problems* (2) *and* (3) *are NP-hard.*

Although the optimization problem (2) is NP-hard in general, but it can be tractable in certain settings. One such setting is when the Markov chain induced by the baseline policy is known precisely, as the following proposition states. See Appendix E for the proof.

Proposition 7. *Assume that for each $x \in \mathcal{X}$, the error function induced by the baseline policy is zero, i.e., $e\left(x, \pi_B(x)\right) = 0$.[2] Then, the optimization problem* (2) *is equivalent to the following robust MDP (RMDP) problem and can be solved in polynomial time:*

$$\arg\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \rho(\pi, \xi). \tag{4}$$

3.4 Approximate Algorithm

Solving for the optimal solution of (2) may not be possible in practice, since the problem is NP hard. In this section, we propose a simple and practical approximate algorithm. The empirical results of Section 5 indicate that this algorithm holds promise and also suggest that the approach may be a good starting point for building better approximate algorithms in the future.

Algorithm 1: Approximate Robust Baseline Regret Minimization Algorithm

---

input : Empirical transition probabilities: $\widehat{P}$, baseline policy π_B, and the error function e
output: Policy $\tilde{\pi}_S$
1 **foreach** $x \in \mathcal{X}, a \in \mathcal{A}$ **do**
2 $\quad$ $\tilde{e}(x, a) \leftarrow \begin{cases} e(x, a) & \text{when } \pi_B(x) \neq a \\ 0 & \text{otherwise} \end{cases}$;
3 **end**
4 $\tilde{\pi}_S \leftarrow \arg\max_{\pi \in \Pi_R} \min_{\xi \in \Xi(\widehat{P}, \tilde{e})} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi)\right)$;
5 **return** $\tilde{\pi}_S$

Algorithm 1 contains the pseudocode of the proposed approximate method. The main idea is to use a modified uncertainty model by assuming no error in transition probabilities of the baseline

---

[2]Note that this is equivalent to precisely knowing the Markov chain induced by the baseline policy $P^\star_{\pi_B}$.

policy. Then it is possible to minimize the robust baseline regret in polynomial time as suggested by Theorem 7. Assuming no error in baseline transition probabilities is reasonable because of two main reasons. First, in practice, data is often generated by executing the baseline policy, and thus, we may have enough data for a good approximation of the baseline's transition probabilities: $\forall x \in \mathcal{X}, \widehat{P}\big(\cdot \,|x, \pi_B(x)\big) \approx P^\star\big(\cdot \,|x, \pi_B(x)\big)$. Second, transition probabilities often affect baseline and improved policies similarly, and as a result, have little effect on the difference between their returns (i.e., the regret). See Section 5.1 for an example of such behavior.

4 Standard Policy Improvement Methods

In Section 3, we showed that finding an exact solution to the optimization problem (2) is computationally expensive and proposed an approximate algorithm. In this section, we describe and analyze two standard methods for computing safe policies and show how they can be interpreted as an approximation of our proposed baseline regret minimization. Due to space limitations, we describe another method, called reward-adjusted MDP, in Appendix H, but report its performance in Section 5.

4.1 Solving the Simulator

The simplest solution to (2) is to assume that our *simulator* is accurate and to solve the reward maximization problem of an MDP with the transition probability $\widehat{P}$, i.e., $\pi_{\text{sim}} \in \arg\max_{\pi \in \Pi_R} \rho(\pi, \widehat{P})$. Theorem 8 quantifies the performance loss of the resulted policy π_{sim}.

Theorem 8. *Let π_{sim} be an optimal policy of the reward maximization problem of an MDP with transition probability $\widehat{P}$. Then under Assumption 1, the performance loss of π_{sim} is bounded by*

$$\Phi(\pi_{sim}) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_{sim}, P^\star) \leq \frac{2\gamma R_{\max}}{(1-\gamma)^2}\|e\|_\infty.$$

The proof is available in Appendix F. Note that there is no guarantee that π_{sim} is *safe*, and thus, deploying it may lead to undesirable outcomes due to model uncertainties. Moreover, the performance guarantee of π_{sim}, reported in Theorem 8, is weaker than that in Theorem 5 due to the L_∞ norm.

4.2 Solving Robust MDP

Another standard solution to the problem in (2) is based on solving the RMDP problem (4). We prove that the policy returned by this algorithm is *safe* and has better (sharper) worst-case guarantees than the simulator-based policy π_{sim}. Details of this algorithm are summarized in Algorithm 2. The algorithm first constructs and solves an RMDP. It then returns the solution policy if its worst-case performance over the uncertainty set is better than the robust performance $\max_{\xi \in \Xi} \rho(\pi_B, \xi)$, and it returns the baseline policy π_B, otherwise.

Algorithm 2: RMDP-based Algorithm

---

input : Simulated MDP $\widehat{P}$, baseline policy π_B, and the error function e
output: Policy π_R

1 $\pi_0 \leftarrow \arg\max_{\pi \in \Pi_R} \min_{\xi \in \Xi(\widehat{P}, e)} \rho(\pi, \xi)$;
2 **if** $\min_{\xi \in \Xi(\widehat{P}, e)} \rho(\pi_0, \xi) > \max_{\xi \in \Xi} \rho(\pi_B, \xi)$ **then return** π_0 **else return** π_B ;

Algorithm 2 makes use of the following approximation to the solution of (2):

$$\max_{\pi \in \Pi_R} \min_{\xi \in \Xi}\Big(\rho(\pi, \xi) - \rho(\pi_B, \xi)\Big) \geq \max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \rho(\pi, \xi) - \max_{\xi \in \Xi} \rho(\pi_B, \xi),$$

and guarantees safety by designing π such that the RHS of this inequality is always non-negative.

The performance bound of π_R is identical to that in Theorem 5 and is stated and proved in Theorem 12 in Appendix G. Although the worst-case bounds are the same, we show in Section 5.1 that the performance loss of π_R may be worse than that of π_S by an arbitrarily large margin.

It is important to discuss the difference between Algorithms 1 and 2. Although both solve an RMDP, they use different uncertainty sets Ξ. The uncertainty set used in Algorithm 2 is the true error function in building the simulator, while the uncertainty set used in Algorithm 1 assumes that the error function is zero for all the actions suggested by the baseline policy. As a result, both algorithms approximately solve (2) but approximate the problem in different ways.

5 Experimental Evaluation

In this section, we experimentally evaluate the benefits of minimizing the robust baseline regret. First, we demonstrate that solving the problem in (2) may outperform the regular robust formulation by an arbitrarily large margin. Then, in the remainder of the section, we compare the solution quality of Algorithm 1 with simpler methods in more complex and realistic experimental domains. The purpose of our experiments is to show how solution quality depends on the degree of model uncertainties.

5.1 An Illustrative Example

Consider the example depicted on the right panel of Figure 1. White nodes represent states and black nodes represent state-action pairs. Labels on the edges originated from states indicate the policy according to which the action is taken; labels on the edges originated from actions denote the rewards and, if necessary, the name of the uncertainty realization. The baseline policy is π_B, the optimal policy is $\pi^\star$, and the discount factor is $\gamma \in (0, 1)$.

This example represents a setting in which the level of uncertainty varies significantly across the individual states: the transition model is precise in state x_0 and uncertain in state x_1. The baseline policy π_B takes a suboptimal action in state x_0 and the optimal action in the uncertain state x_1. To prevent being overly conservative in computing a safe policy, one needs to consider that the realization of uncertainty in x_1 influences both the baseline and improved policies.

Using the plain robust optimization formulation in Algorithm 2, even the optimal policy $\pi^\star$ is not considered safe in this example. In particular, the robust return of $\pi^\star$ is $\min_\xi \rho(\pi^\star, \xi) = -9$, while the optimistic return of π_B is $\max_\xi \rho(\pi_B, \xi) = +10$. On the other hand, solving (2) will return the optimal policy since: $\min_\xi \rho(\pi^\star, \xi) - \rho(\pi_B, \xi) = 11 - 10 = -9 - (-10) = 1$. Even the heuristic method of Section 3.4 will return the optimal policy. Note that since the reward-adjusted formulation (see its description in Appendix H) is even more conservative than the robust formulation, it will also fail to improve on the baseline policy.

5.2 Grid Problem

In this section, we use a simple grid problem to compare the solution quality of Algorithm 1 with simpler methods. The grid problem is motivated by modeling customer interactions with an online system. States in the problem represent a two dimensional grid. Columns capture states of interaction with the website and rows capture customer states such as overall satisfaction. Actions can move customers along either dimension with some probability of failure. A more detailed description of this domain is provided in Section I.1.

Our goal is to evaluate how the solution quality of various methods depends on the magnitude of the model error e. The model is constructed from samples, and thus, its magnitude of error depends on the number of samples used to build it. We use a uniform random policy to gather samples. Model error function e is then constructed from this simulated data using bounds in Section B. The baseline policy is constructed to be optimal when ignoring the row part of state; see Section I.1 for more details.

All methods are compared in terms of the improvement percentage in total return over the baseline policy. Figure 2 depicts the results as a function of the number of transition samples used in constructing the uncertain model and represents the mean of 40 runs. Methods used in the comparison are as follows: **1)** EXP represents solving the nominal model as described in Section 4.1, **2)** RWA represent the reward-adjusted formulation in Algorithm 3 of Appendix H, **3)** ROB represents the robust method in Algorithm 2, and **4)** RBC represents our approximate solution of Algorithm 1.

Figure 2 shows that Algorithm 1 not only reliably computes policies that are safe, but also significantly improves on the quality of the baseline policy when the model error is large. When the number of
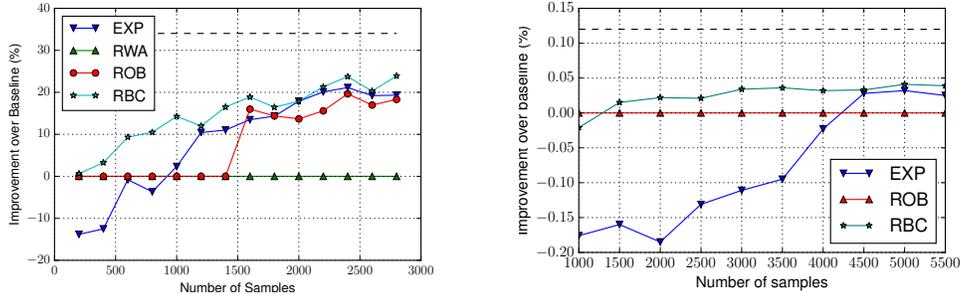
Figure 2: Improvement in return over the baseline policy in: (left) the grid problem and (right) the energy arbitrage problem. The dashed line shows the return of the optimal policy.

samples is small, Algorithm 1 is significantly better than other methods by relying on the baseline policy in states with a large model error and only taking improving actions when the model error is small. Note that EXP can be significantly worse than the baseline policy, especially when the number of samples is small.

5.3 Energy Arbitrage

In this section, we compare model-based policy improvement methods using a more complex domain. The problem is to determine an energy arbitrage policy in given limited energy storage (a battery) and stochastic prices. At each time period, the decision-maker observes the available battery charge and a Markov state of energy price, and decides on the amount of energy to purchase or to sell.

The set of states in the energy arbitrage problem consists of three components: current state of charge, current capacity, and a Markov state representing price; the actions represent the amount of energy purchased or sold; the rewards indicate profit/loss in the transactions. We discretize the state of charge and action sets to 10 separate levels. The problem is based on the domain from [Petrik and Wu, 2015], whose description is detailed in Appendix I.2.

Energy arbitrage is a good fit for model-based approaches because it combines known and unknown dynamics. Physics of battery charging and discharging can be modeled with high confidence, while the evolution of energy prices is uncertain. As a result, using an explicit battery model, the only uncertainty is in transition probabilities between the 10 states of the price process instead of the entire 1000 state-action pairs. This significantly reduces the number of samples needed to compute a good solution.

As in the previous experiments, we estimate the uncertainty model in a data-driven manner. Notice that the inherent uncertainty is only in price transitions and is independent of the policy used (which controls the storage dynamics). Here the uncertainty set of transition probabilities is estimated using the method in Appendix A, but the uncertainty set is only a non-singleton w.r.t. price states. Figure 2 shows the percentage improvement on the baseline policy averaged over 5 runs. We clearly observe that the heuristic RBC method, described in Section 3.4, effectively interleaves the baseline policy (in states with high level of uncertainty) and an improved policy (in states with low level of uncertainty), and results in the best performance in most cases.

6 Conclusion

In this paper, we studied the *model-based* approach to the fundamental problem of learning *safe* policies given a batch of data. A policy is considered *safe*, if it is guaranteed to have an improved performance over a baseline policy. Solving the problem of safety in sequential decision-making can immensely increase the applicability of the existing technology to real-world problems. We showed that the standard robust formulation may be overly conservative and formulated a better approach that interleaves an improved policy with the baseline policy, based on the error at each state. We proposed and analyzed an optimization problem based on this idea (see (2)) and proved that solving it is NP-hard. Furthermore, we proposed several approximate solutions and experimentally evaluated their performances.

References

A. Ahmed and P Varakantham. Regret based Robust Solutions for Uncertain Markov Decision Processes. *Advances in neural information processing systems*, pages 1–9, 2013.

T. Hansen, P. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM*, 60(1):1–16, 2013.

G. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, pages 267–274, 2002.

Y. Le Tallec. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes*. PhD thesis, MIT, 2007.

A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

M. Petrik and D. Subramanian. RAAM : The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Neural Information Processing Systems*, 2014.

M. Petrik and X. Wu. Optimal Threshold Control for Energy Arbitrage with Degradable Battery Storage. In *Uncertainty in Artificial Intelligence*, pages 692–701, 2015.

M. Pirotta, M. Restelli, and D. Calandriello. Safe Policy Iteration. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

P. Thomas, G. Teocharous, and M. Ghavamzadeh. High Confidence Policy Improvement. In *International Conference on Machine Learning*, pages 2380–2388, 2015.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*, 2015.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. Weinberger. Inequalities for the L_1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

W. Wiesemann, D. Kuhn, and B. Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

H. Xu and S. Mannor. Parametric regret in uncertain Markov decision processes. *Proceedings of the IEEE Conference on Decision and Control*, pages 3606–3613, 2009.

A Error Bound

Our goal here is to construct the error function e, when $\widehat{P}$ is estimated from the samples drawn from $P^\star$, such that we can guarantee that $P^\star \in \Xi(\widehat{P}, e)$, with probability at least $1 - \delta$. Let us assume that at each state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$, we draw $N(x, a)$ samples from $P^\star(\cdot|x, a)$.

Proposition 9. *If at each state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$, we define $e(x, a) = \sqrt{\frac{2}{N(x,a)} \log \left(\frac{|\mathcal{X}||\mathcal{A}|2^{|\mathcal{X}|}}{\delta} \right)}$, then $P^\star \in \Xi(\widehat{P}, e)$, with probability at least $1 - \delta$.*

Proof. From Theorem 2.1 in Weissman *et al.* [2003], for each state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$, we may write

$$\mathbb{P}\left(||P^\star(\cdot \mid x, a) - \widehat{P}(\cdot \mid x, a)||_1 \geq \epsilon \right) \leq (2^{|\mathcal{X}|} - 2) \exp\left(-\frac{N(x, a)\epsilon^2}{2} \right). \tag{5}$$

Setting $\epsilon = \sqrt{\frac{2}{N(x,a)} \log \left(\frac{|\mathcal{X}||\mathcal{A}|2^{|\mathcal{X}|}}{\delta} \right)}$, we may rewrite (5) as

$$\mathbb{P}\Big(||P^\star(\cdot \mid x, a) - \widehat{P}(\cdot \mid x, a)||_1 \geq \sqrt{\frac{2}{N(x, a)} \log \left(\frac{|\mathcal{X}||\mathcal{A}|2^{|\mathcal{X}|}}{\delta} \right)} \Big)$$

$$\leq 2^{|\mathcal{X}|} \exp\left(-\frac{N(x, a)}{2} \times \frac{2}{N(x, a)} \log \left(\frac{|\mathcal{X}||\mathcal{A}|2^{|\mathcal{X}|}}{\delta} \right) \right)$$

$$= \frac{\delta}{|\mathcal{X}||\mathcal{A}|}. \tag{6}$$

From the definition of the uncertainty set $\Xi(\widehat{P}, e)$ and by summing the error probability in (6), we obtain that $\mathbb{P}\big(P^\star \notin \Xi(\widehat{P}, e) \big) \leq \delta$. $\qquad\square$

B   Proof of Lemma 11

for which the following technical lemma (whose proof can be found in Appendix B) is used in the analysis.

Before proving Lemma 11, we first prove the following lemma.

Lemma 10. *For any policy $\pi \in \Pi_R$, consider two transition probability matrices P_1 and P_2 and two reward functions r_1 and r_2 corresponding to π. Let v_1 and v_2 be the value functions of the policy π given (P_1, r_1) and (P_2, r_2), respectively. Under the assumption that for any state $x \in \mathcal{X}$, $\|P_1(\cdot|x) - P_2(\cdot|x)\|_1 \le e(x)$, we have*

$$(\mathbf{I} - \gamma P_1)^{-1}\left(r_1 - r_2 - \frac{\gamma R_{\max}}{1 - \gamma}e\right) \le v_1 - v_2 \le (\mathbf{I} - \gamma P_1)^{-1}\left(r_1 - r_2 + \frac{\gamma R_{\max}}{1 - \gamma}e\right),$$

where e is the vector of $e(x)$'s.

Proof. The difference between the two value functions can be written as

$$\begin{aligned}
v_1 - v_2 &= r_1 + \gamma P_1 v_1 - r_2 - \gamma P_2 v_2 \\
&= r_1 + \gamma P_1 v_1 - r_2 - \gamma P_2 v_2 + \gamma P_1 v_2 - \gamma P_1 v_2 \\
&= (r_1 - r_2) + \gamma P_1 (v_1 - v_2) + \gamma (P_1 - P_2) v_2 \\
&= (\mathbf{I} - \gamma P_1)^{-1}\left[r_1 - r_2 + \gamma(P_1 - P_2)v_2\right].
\end{aligned}$$

Now using the Holder's inequality, for any $x \in \mathcal{X}$, we have

$$\left|\left(P_1(\cdot|x) - P_2(\cdot|x)\right)^\mathsf{T} v_2\right| \le \|P_1(\cdot|x) - P_2(\cdot|x)\|_1 \|v_2\|_\infty \le e(x)\|v_2\|_\infty \le e(x)\frac{R_{\max}}{1 - \gamma}.$$

The proof follows by uniformly bounding $(P_1 - P_2)v_2$ from the above inequality and from the monotonicity of $(\mathbf{I} - \gamma P_1)^{-1}$. □

Lemma 11. *The difference between the returns of a policy π in two MDPs parameterized by $P^\star, \xi \in \Xi$ is bounded as*

$$|\rho(\pi, P^\star) - \rho(\pi, \xi)| \le \frac{2\gamma R_{\max}}{1 - \gamma} \, p_0^\mathsf{T}(\mathbf{I} - \gamma P_\pi^\star)^{-1}e_\pi,$$

where $P_\pi^\star$ and e_π are the transition probability matrix and error function (between $P^\star$ and ξ, see Eq. 1) of policy π.

Proof. Lemma 11 is the direct consequence of Lemma 10 with the fact that for any $(x, a) \in \mathcal{X} \times \mathcal{A}$ and any $\xi \in \Xi(\widehat{P}, e)$, from Assumption 1 and the construction of $\Xi(\widehat{P}, e)$, we have

$$\begin{aligned}
\|P^\star(\cdot|x, a) - \xi(\cdot|x, a)\|_1 &= \|P^\star(\cdot|x, a) - \widehat{P}(\cdot|x, a) + \widehat{P}(\cdot|x, a) - \xi(\cdot|x, a)\|_1 \\
&\le \|P^\star(\cdot|x, a) - \widehat{P}(\cdot|x, a)\|_1 + \|\widehat{P}(\cdot|x, a) - \xi(\cdot|x, a)\|_1 \\
&\le 2e(x, a) \, .
\end{aligned}$$

□

C Proof of Theorem 5

To prove the safety of π_S, note that the objective in (2) is always non-negative, since the baseline policy π_B is a feasible solution. Thus, we obtain the safety condition by simple algebraic manipulation as follows:

$$\rho(\pi_S, P^\star) - \rho(\pi_B, P^\star) \geq \min_{\xi \in \Xi} \Big(\rho(\pi_S, \xi) - \rho(\pi_B, \xi) \Big) = \max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \Big(\rho(\pi, \xi) - \rho(\pi_B, \xi) \Big) \geq 0 . \quad (7)$$

Now we prove the performance bound. From Theorem 11, for any policy π, we may write

$$\max_\xi \Big| \rho(\pi, \xi) - \rho(\pi, P^\star) \Big| \leq \frac{2\gamma R_{\max}}{1 - \gamma} p_0^\mathsf{T} (\mathbf{I} - P_\pi^\star)^{-1} e_\pi = \frac{2\gamma R_{\max}}{(1 - \gamma)^2} \|e_\pi\|_{1, u_\pi^\star} , \quad (8)$$

where $u_\pi^\star$ is state occupancy distribution of policy π in the true MDP $P^\star$, defined as

$$u_\pi^\star = (1 - \gamma)(\mathbf{I} - \gamma P_\pi^{\star\top})^{-1} p_0.$$

We are now ready to show a bound on the performance loss of π_S through the following set of inequalities:

$$\Phi(\pi_S) = \rho(\pi^\star, P^\star) - \rho(\pi_S, P^\star) = \rho(\pi^\star, P^\star) - \rho(\pi_S, P^\star) + \rho(\pi_B, P^\star) - \rho(\pi_B, P^\star)$$

$$\leq \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star) - \min_\xi \Big(\rho(\pi_S, \xi) - \rho(\pi_B, \xi) \Big)$$

$$\leq \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star) - \min_\xi \Big(\rho(\pi^\star, \xi) - \rho(\pi_B, \xi) \Big)$$

$$\leq \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star) - \min_\xi \rho(\pi^\star, \xi) + \max_\xi \rho(\pi_B, \xi)$$

$$= \max_\xi \Big(\rho(\pi^\star, P^\star) - \rho(\pi^\star, \xi) \Big) + \max_\xi \Big(\rho(\pi_B, \xi) - \rho(\pi_B, P^\star) \Big)$$

$$\overset{(a)}{\leq} \frac{2\gamma R_{\max}}{(1 - \gamma)^2} \Big(\|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star} + \|e_{\pi_B}\|_{1, u_{\pi_B}^\star} \Big) , \quad (9)$$

where **(a)** is by applying (8) to the two $\max$ terms on the RHS of the inequality.

The final bound is obtained by combining (9) and the fact that $\rho(\pi_S, P^\star) \geq \rho(\pi_B, P^\star)$, and as a result, $\Phi(\pi_S) \leq \Phi(\pi_B)$.

To prove the tightness of the bound, we use the example depicted in Figure 3. The initial state is x_0, actions are a_1, a_2, the transitions are deterministic, and the leafs represent absorbing states with the given return. We denote by $P^\star$, the transitions of the true MDP, and by ξ_1, the worst-case transitions in the uncertainty set $\Xi(\widehat{P}, e)$. Finally the baseline policy π_B takes action a_1 in state x_0 and shown by double edges in Figure 3. It is clear that the optimal policy $\pi^\star$ is the one that takes action a_2 in state x_0. The return of this policy is $\rho(\pi^\star, P^\star) = 1 + 2\epsilon$. It is also straightforward to derive that the policy π_S that takes action a_1 in state x_0 (as shown in Figure 3) is a solution to (2). The return of this policy is $\rho(\pi_S, P^\star) = 1$ and its performance loss is $\Phi(\pi_S) = \rho(\pi^\star, P^\star) - \rho(\pi_S, P^\star) = 2\epsilon$.
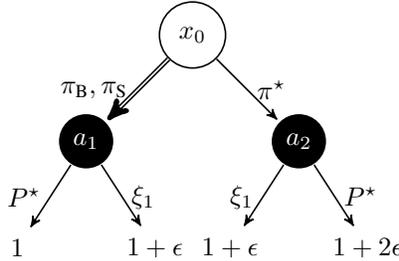

Figure 3: Example showing the tightness of the bound in Theorem 5.

Now let us set ϵ in the leafs of Figure 3 to $\epsilon = \frac{2\gamma R_{\max}}{(1-\gamma)^2} \|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star}$. Note that this is the value given by (8) for $\pi = \pi^\star$. This gives us the tightness proof assuming that $\widehat{P}$ is such that $\|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star}$ and $\|e_{\pi_B}\|_{1, u_{\pi_B}^\star}$ have similar values, and $1 + 2\epsilon$ is a valid return value, i.e., $1 + 2\epsilon \leq \frac{R_{\max}}{1-\gamma}$.
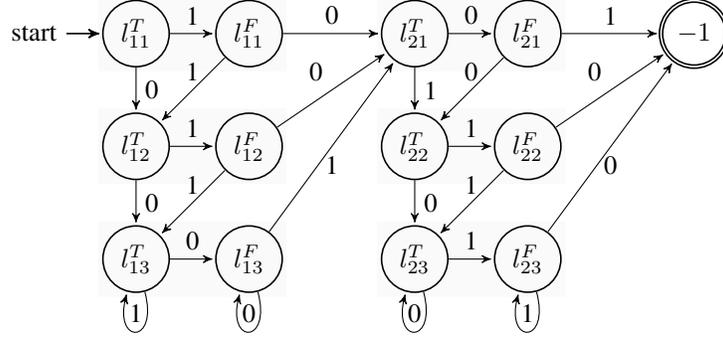
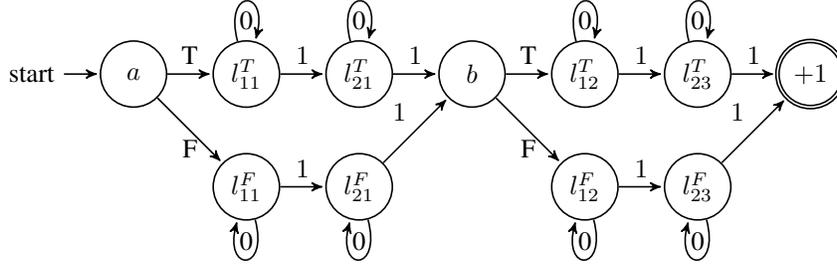Figure 4: MDP $\mathcal{M}_1$ in Theorem 6 that represents the optimization of $\rho(\pi, \xi)$ over ξ.

Figure 5: MDP $\mathcal{M}_2$ in Theorem 6 representing the optimization of $\rho(\pi_B, \xi)$ over ξ.

D Proof of Theorem 6

Assume a given fixed policy π. We start by showing the NP hardness of solving (3):

$$\min_{\xi \in \Xi} \left(\rho(\pi, \xi) - \rho(\pi_B, \xi) \right)$$

by a reduction from the boolean satisfiability (SAT) problem. To simplify the exposition, we also illustrate the reduction on the following simple example SAT problem in a conjunctive normal form (CNF):

$$(a \vee b \vee \neg c) \wedge (\neg a \vee d \vee b) = (l_{11} \vee l_{12} \vee l_{13}) \wedge (l_{21} \vee l_{22} \vee l_{23}) , \tag{10}$$

where a, b, c, and d are the variables, and l_{ij} represent the j-th literal in i-th disjunction.

As noted above, $\rho(\pi, \xi)$ represents the return of a robust MDP. Recall that computing $\min_\xi \rho(\pi, \xi)$ for a fixed π is equivalent to computing a policy in a regular MDP with actions representing realizations of the transition uncertainty. Therefore, optimizing for ξ in (3) translates to finding a single policy ξ for two MDPs—defined by π and π_B—that maximizes the difference between their returns $\rho(\pi, \xi) - \rho(\pi_B, \xi)$.

We reduce the SAT problem to the optimization over ξ in (3). As described above, the value $\rho(\pi, \xi)$ for a fixed π can be represented as a return of some MDP $\mathcal{M}_1$ for a policy given by ξ. Similarly, the value $\rho(\pi_B, \xi)$ for a fixed π_B can be represented as a return of another MDP $\mathcal{M}2$. We describe the general reduction in detail below. Figures 4 and 5 illustrate the MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively for the example in (10).

MDPs $\mathcal{M}_1, \mathcal{M}_2$ share the same state and action sets. The actions represent the realization of uncertainty ξ and are denoted by the edge labels. They are discrete and stand for the extreme points of feasible ℓ_1 uncertainty sets. For ease of notation, we assume $\gamma = 1$ and states with double circles are terminal with rewards inscribed therein. All non-terminal transition have zero rewards.

The identical state set of both $\mathcal{M}_1$ and $\mathcal{M}_2$ are constructed as follows. There is one state for each variable $v \in \{a, b, c, d\}$, and two states $\{l_{ij}^T, l_{ij}^F\}$ for every literal l_{ij}. Informally, actions $\{T, F\}$ for a variable state capture the value of that variable. Actions $\{0, 1\}$ for a literal state l_{ij}^T or l_{ij}^F represent

13

the value of the variable referenced by the literal. This is regardless of whether the literal is positive or negative. For example, when the variable in l_{ij} is true, the action in l_{ij}^T is 1 and when the variable in l_{ij} is false, the action in l_{ij}^F is 1. Two states per each literal are necessary in order to model the negation operation.

The transitions in MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ are constructed to guarantee that their returns are -1 and $+1$, respectively (and as a result the objective in (3) is -2), only if the assignment to the literals satisfies the SAT problem. Note that the transitions for the negated literals, such as l_{21} are different from the positive literals, such as l_{11}. This construction easily generalizes to any SAT problem in the CNF. Consider the example in (10) and let $b = T$ (other variables can take any values). It can then be seen readily that the objective in (3) would be -2.

Let $\rho^\star$ be the optimal value of (3). Then, to show the correctness of our reduction, we argue that $\rho^\star = -2$, if and only if the SAT problem is satisfiable. To show the reverse implication, assume that the SAT is satisfied for some assignment to variables and construct a policy $\bar{\xi}$ as follows:

$$\bar{\xi}(v) = \begin{cases} T & \text{if } v = \text{true} \\ F & \text{otherwise} \end{cases}, \qquad \bar{\xi}(l_{ij}^T) = \begin{cases} 1 & \text{if } v_{ij} = \text{true} \\ 0 & \text{otherwise} \end{cases}, \qquad \bar{\xi}(l_{ij}^F) = \begin{cases} 0 & \text{if } v_{ij} = \text{true} \\ 1 & \text{otherwise} \end{cases},$$

where v_{ij} represents the value of the variable referenced by the corresponding literal l_{ij}, e.g., $v_{11} = v_{21} = a$ in (10). It can be readily seen that $\rho(\pi, \bar{\xi}) = 1$ and $\rho(\pi_B, \bar{\xi}) = -1$, and thus, the implication that $\rho^\star = -2$ holds.

To show the forward implication, assume that for an optimal deterministic realization $\bar{\xi}$, we have $\rho(\pi_B, \xi^\star) = 1$ and $\rho(\pi_B, \xi^\star) = -1$, and thus, $\rho^\star = -2$. We assign values to variables v as follows:

$$v = \begin{cases} \text{true} & \text{if } \bar{\xi}(v) = T , \\ \text{false} & \text{otherwise} . \end{cases}$$

We have that $\rho(\pi_B, \bar{\xi}) = 1$ only if for every disjunction i either **1)** there exists a positive literal l_{ij} such that $\bar{\xi}(l_{ij}^T) = 1$ and $\bar{\xi}(l_{ij}^F) = 0$, or **2)** there exists a negative literal l_{ij} such that $\bar{\xi}(l_{ij}^T) = 0$ and $\bar{\xi}(l_{ij}^F) = 1$. Assume without loss of generality that this is always the first literal l_{i1}. Now, consider any positive $l_{i1} = v$ and observe that $\bar{\xi}(l_{i1}^T) = 1$ and $\bar{\xi}(l_{i1}^F) = 0$. Because $\rho(\pi_B, \bar{\xi}) = 1$ only if $\bar{\xi}(v) = T$, the disjunction i is satisfied. The case of a negative l_{i1} is analogous, and thus, the forward implication also holds.

The restriction to deterministic policies $\bar{\xi}$ in the forward implication argument can be lifted by considering a discount factor; in such case the maximal return in $\mathcal{M}_2$ may be achieved only by a deterministic policy. Then, appropriately increasing the return in $\mathcal{M}_2$ finishes the argument.

The argument above shows that the inner minimization problem in (2) is NP hard. Recall that (2) is stated as follows:

$$\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \big(\rho(\pi, \xi) - \rho(\pi_B, \xi)\big)$$

To prove the theorem, it simply remains to show that the outer maximization over π does not make the problem any easier. To show this, we will construct a single robust MDP $\mathcal{R}$ such that a policy with the maximal improvement induces $\mathcal{M}_1$ as the robust optimization subproblem. Baseline policy π_B in $\mathcal{R}$ similarly induces $\mathcal{M}_2$. Then, the difference between improved and baseline policies is no greater than some threshold if and only if the SAT is satisfiable.

Construct the robust MDP $\mathcal{R}$ with the same state set as $\mathcal{M}_1$ and $\mathcal{M}_2$. There are two actions a_1 and a_2 in each state. Upon taking action a_1, the transitions are chosen according to $\mathcal{M}_1$ and the reward is as in $\mathcal{M}_1$. Upon taking action a_2, the transition and reward are given the same as in $\mathcal{M}_2$ minus 3. Rewards in terminal states are not modified.

The baseline policy takes action a_2, i.e. $\pi_B(x) = a_2$. Return of the baseline policy is in $[3\,k, 3\,k + 1]$ where k is the sum of the number of distinct variables and literals in the CNF.

Let the improvement policy π' be $\pi'(x) = a_1$. It can be readily seen that this policy achieves the maximal improvement. This is because $\rho(\pi', \xi) \in [0, -1]$ while the return of any other policy will be at most -3 (the return for a_2 in any state is -3).

To finish the proof, observe that when the SAT is satisfiable then:

$$\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \big(\rho(\pi, \xi) - \rho(\pi_B, \xi)\big) = \min_{\xi \in \Xi} \big(\rho(\pi', \xi) - \rho(\pi_B, \xi)\big) = 3\,k - 2 .$$

This is true using the above argument concerning the optimal value of the inner minimization problem. On the other hand, when the SAT is unsatisfiable then by the same argument:

$$\max_{\pi \in \Pi_R} \min_{\xi \in \Xi} \big(\rho(\pi, \xi) - \rho(\pi_{\mathrm{B}}, \xi)\big) = \min_{\xi \in \Xi} \big(\rho(\pi', \xi) - \rho(\pi_{\mathrm{B}}, \xi)\big) > 3\,k - 2\,.$$

This shows that deciding whether the optimal value of (2) is greater than $3\,k - 2$ is as hard as solving the corresponding SAT.

E Proof of Proposition 7

The hypothesis in the proposition implies that for any $\xi \in \Xi(\widehat{P}, e)$, we have $\xi(\,\cdot\,|x, \pi_B(x)) = \widehat{P}(\,\cdot\,|x, \pi_B(x))$, $\forall x \in \mathcal{X}$. This further indicates that $\rho(\pi_B, \xi)$ is a constant (independent of ξ), for all $\xi \in \Xi(\widehat{P}, e)$. Thus, when the Markov chain induced by the baseline policy is known, the optimization problem (2) is reduced to the optimization problem (4), which is a robust MDP (RMDP) problem with ℓ_1-constraint uncertainty set. It is known that this class of RMDP problems can be solved in (strongly) polynomial time [Hansen *et al.*, 2013] and has also been solved efficiently in practice [Petrik and Subramanian, 2014].

F Proof of Theorem 8

From Lemma 11 with $\pi = \pi_{\text{sim}}$ and $\xi = \widehat{P}$ we have

$$\rho(\pi_{\text{sim}}, \widehat{P}) - \frac{\gamma R_{\max}}{1-\gamma} p_0^\top (\mathbf{I} - \gamma P_{\pi_{\text{sim}}}^\star)^{-1} e_{\pi_{\text{sim}}} \le \rho(\pi_{\text{sim}}, P^\star).$$

Thus, we may write

$$\Phi(\pi_{\text{sim}}) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_{\text{sim}}, P^\star) \le \rho(\pi^\star, P^\star) - \rho(\pi_{\text{sim}}, \widehat{P}) + \frac{\gamma R_{\max}}{1-\gamma} p_0^\top (\mathbf{I} - \gamma P_{\pi_{\text{sim}}}^\star)^{-1} e_{\pi_{\text{sim}}}$$

$$\overset{(a)}{\le} \rho(\pi^\star, P^\star) - \rho(\pi^\star, \widehat{P}) + \frac{\gamma R_{\max}}{1-\gamma} p_0^\top (\mathbf{I} - \gamma P_{\pi_{\text{sim}}}^\star)^{-1} e_{\pi_{\text{sim}}}$$

$$\overset{(b)}{\le} \frac{\gamma R_{\max}}{1-\gamma} p_0^\top \left[(\mathbf{I} - \gamma P_{\pi^\star}^\star)^{-1} e_{\pi^\star} + (\mathbf{I} - \gamma P_{\pi_{\text{sim}}}^\star)^{-1} e_{\pi_{\text{sim}}}\right]$$

$$\overset{(c)}{\le} \frac{2\gamma R_{\max}}{(1-\gamma)^2} \|e\|_\infty \,,$$

where each step follows because:

(a) Optimality of π_{sim} in the MDP with transition probabilities $\widehat{P}$.

(b) Application of Lemma 11 with policy $\pi = \pi^\star$ and $\xi = \widehat{P}$.

(c) For any policy $\pi \in \Pi_R$, we have that $\|p_0^\top (\mathbf{I} - \gamma P_\pi^\star)^{-1}\|_1 = 1/(1-\gamma)$, and from the application of the Holder's inequality.

G   Performance Bound on the Solution of the Robust Algorithm

Theorem 12. *Given Assumption 1, the nonempty solution π_R of Algorithm 2 is safe, i.e., $\rho(\pi_R, P^\star) \geq \rho(\pi_B, P^\star)$. Moreover, its performance loss $\Phi(\pi_R)$ satisfies*

$$\Phi(\pi_R) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_R, P^\star) \leq \min\left\{ \frac{2\gamma R_{\max}}{(1-\gamma)^2}\left(\|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star} + \|e_{\pi_B}\|_{1, u_{\pi_B}^\star}\right), \Phi(\pi_B)\right\},$$

where $u_{\pi^\star}^\star$ is the state occupancy distribution of the optimal policy $\pi^\star$ in the true MDP $P^\star$, and $\Phi(\pi_B) = \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star)$ is the performance loss of the baseline policy.

Proof. To prove the safety of π_R and bound its performance loss, we need to upper and lower bound the difference between the performance of any policy π in the true MDP $P^\star$ and its worst-case performance in the uncertainty set Ξ, i.e., $\min_{\xi \in \Xi} \rho(\pi, \xi)$. Since from Assumption 1, we have $P^\star \in \Xi$, we may write

$$\min_{\xi \in \Xi} \rho(\pi, \xi) \leq \rho(\pi, P^\star). \tag{11}$$

Now let $\bar{\xi} \in \Xi(\widehat{P}, e)$ be the minimizer in $\min_{\xi \in \Xi} \rho(\pi, \xi)$. The minimizer exists because of the continuity and compactness of the uncertainty set. Applying Lemma 11 with $\xi = \bar{\xi}$, for any policy $\pi \in \Pi_R$, we obtain

$$\rho(\pi, P^\star) - \rho(\pi, \bar{\xi}) = \rho(\pi, P^\star) - \min_{\xi \in \Xi} \rho(\pi, \xi) \leq \frac{2\gamma R_{\max}}{1 - \gamma} p_0^\top (\mathbf{I} - \gamma P_\pi^\star)^{-1} e_\pi = \frac{2\gamma R_{\max}}{(1-\gamma)^2}\|e_\pi\|_{1, u_\pi^\star},$$
$$\tag{12}$$

where $u_\pi^\star = (1-\gamma)p_0^\top(\mathbf{I} - \gamma P_\pi^\star)^{-1}$ is the state occupancy distribution of policy π in the true MDP $P^\star$.

To prove the safety of the returned policy π_R: Consider the two cases on Line 2 of Algorithm 2. When the condition is satisfied, i.e., $\rho_0 > \max_{\xi \in \Xi} \rho(\pi_B, \xi)$, we have

$$\rho(\pi_B, P^\star) \leq \max_{\xi \in \Xi} \rho(\pi_B, \xi) < \underbrace{\min_{\xi \in \Xi} \rho(\pi_0, \xi)}_{\rho_0} \leq \rho(\pi_0, P^\star),$$

where the last inequality comes from (11), and thus, the policy $\pi_R = \pi_0$ is safe. When the condition is violated, then π_R is simply π_B, which is safe by definition.

To derive a bound on the performance loss of the returned policy π_R: Consider also the two cases on Line 2 of Algorithm 2. When the condition is satisfied, using (11), we have

$$\Phi(\pi_R) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_R, P^\star) = \rho(\pi^\star, P^\star) - \rho(\pi_0, P^\star) \leq \rho(\pi^\star, P^\star) - \min_{\xi \in \Xi} \rho(\pi_0, \xi),$$

and when the condition is violated, we have

$$\Phi(\pi_R) \triangleq \rho(\pi^\star, P^\star) - \rho(\pi_R, P^\star) = \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star).$$

Since when the condition is satisfied on Line 2 of Algorithm 2, we have

$$\min_{\xi \in \Xi} \rho(\pi_0, \xi) > \max_{\xi \in \Xi} \rho(\pi_B, \xi)$$

in both cases on Line 2 of Algorithm 2, we may write

$$\Phi(\pi_R) \leq \min\left\{ \rho(\pi^\star, P^\star) - \min_{\xi \in \Xi} \rho(\pi_0, \xi) + \max_{\xi \in \Xi} \rho(\pi_B, \xi) - \rho(\pi_B, P^\star), \overbrace{\rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star)}^{\Phi(\pi_B)}\right\}.$$

The first term in the minimum can be written as

$$\rho(\pi^\star, P^\star) - \min_{\xi \in \Xi} \rho(\pi_0, \xi) + \max_{\xi \in \Xi} \rho(\pi_B, \xi) - \rho(\pi_B, P^\star)$$

$$\overset{(a)}{\leq} \rho(\pi^\star, P^\star) - \min_{\xi \in \Xi} \rho(\pi^\star, \xi) + \max_{\xi \in \Xi} \rho(\pi_B, \xi) - \rho(\pi_B, P^\star)$$

$$\overset{(b)}{\leq} \frac{2\gamma R_{\max}}{(1-\gamma)^2}\|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star} + \frac{2\gamma R_{\max}}{(1-\gamma)^2}\|e_{\pi_B}\|_{1, u_{\pi_B}^\star},$$

where **(a)** follows from π_0 being the solution to (2), and thus, being the maximizer of $\min_{\xi \in \Xi} \rho(\pi, \xi)$, and **(b)** is from (12) with $\pi = \pi^\star$ and $\pi = \pi_B$. $\qquad\square$

H Solving the Reward-Adjusted MDP

In this section, we describe and analyze another simple method for computing safe policies that we did not include it in Section 4 due to space limitations, and show how it can be interpreted as an approximation of our proposed baseline regret minimization. This algorithm is based on solving a MDP with the same transition probabilities as the simulator, $\widehat{P}$, and rewards defined as $\widehat{r}(x,a) = r(x,a) - \frac{\gamma R_{\max}}{1-\gamma}e(x,a), \; \forall(x,a) \in \mathcal{X} \times \mathcal{A}$. We call this MDP, *reward-adjusted* (RaMDP), and denote its transition probabilities and rewards by $\widetilde{\xi}$. The unique property of RaMDP is that under Assumption 1, the performance of any policy π in RaMDP is a lower-bound on its performance in the true MDP, i.e., $\rho(\pi,\widetilde{\xi}) \leq \rho(\pi,P^\star)$ (see Theorem 14). Furthermore in comparison to the objective function of RMDP, the following proposition shows that $\rho(\pi,\widetilde{\xi})$ is always a lower-bound on $\min_{\xi \in \Xi}\rho(\pi,\xi)$.

Proposition 13. *Given Assumption 1, for each policy π, we have $\min_{\xi \in \Xi}\rho(\pi,\xi) \geq \rho(\pi,\widetilde{\xi})$.*

Proof. Let $\bar{\xi} \in \Xi(\widehat{P},e)$ be the minimizer in $\min_{\xi \in \Xi}\rho(\pi,\xi)$. The minimizer exists because of the continuity and compactness of the uncertainty set. From Lemma 10, for each π, we may write

$$\rho(\pi,\bar{\xi}) \geq \rho(\pi,\widehat{P}) - \frac{\gamma R_{\max}}{1-\gamma}p_0^\mathsf{T}(\mathbf{I} - \gamma\widehat{P}_\pi)^{-1}e_\pi \overset{(a)}{=} \rho(\pi,\widetilde{\xi}),$$

where **(a)** holds because $\widetilde{\xi}$ differs from $\widehat{P}$ only in its reward function, which is of the form $\widehat{r}_\pi = r_\pi - \frac{\gamma R_{\max}}{1-\gamma}e_\pi$. $\qquad\square$

We conclude based on this proposition that the reward-adjusted method approximates the solution of the optimization problem (2) as

$$\max_{\pi \in \Pi_R}\min_{\xi \in \Xi}\Big(\rho(\pi,\xi) - \rho(\pi_B,\xi)\Big) \geq \max_{\pi \in \Pi_R}\min_{\xi \in \Xi}\rho(\pi,\xi) - \max_{\xi \in \Xi}\rho(\pi_B,\xi)$$

$$\geq \max_{\pi \in \Pi_R}\rho(\pi,\widetilde{\xi}) - \max_{\xi \in \Xi}\rho(\pi_B,\xi), \tag{13}$$

and guarantees safety by designing π such that the RHS of (13) is always non-negative. Algorithm 3 returns an optimal policy of the RaMDP $\widetilde{\xi}$, when the performance of this policy in $\widetilde{\xi}$ is better than the robust baseline performance $\max_{\xi \in \Xi}\rho(\pi_B,\xi)$, and returns π_B, otherwise.

Algorithm 3: RaMDP-based Algorithm

    input   : Simulated MDP $\widehat{P}$, baseline policy π_B, and the error function e
    output: Policy π_{Ra}
1 $\widehat{r}(x,a) \leftarrow r(x,a) - \frac{\gamma R_{\max}}{1-\gamma}e(x,a)$;
2 $\pi_0 \leftarrow \arg\max_{\pi \in \Pi_R}\rho(\pi,\widetilde{\xi})$;   where $\widetilde{\xi} = (\widehat{r},\widehat{P})$
3 $\rho_0 \leftarrow \rho(\pi_0,\widetilde{\xi})$;
4 **if** $\rho_0 > \max_{\xi \in \Xi}\rho(\pi_B,\xi)$ **then** $\pi_{Ra} \leftarrow \pi_0$ **else** $\pi_{Ra} \leftarrow \pi_B$;
5 **return** π_{Ra}

Since the performance of any policy in the RaMDP $\widetilde{\xi}$ is a lower-bound on its performance in the true MDP $P^\star$, it is guaranteed that the policy π_{Ra} returned by Algorithm 3 performs at least as well as the baseline policy π_B. Theorem 14 shows that π_{Ra} is *safe* and quantifies its performance loss.

Theorem 14. *Given Assumption 1, the solution π_{Ra} of Algorithm 3 is safe, i.e., $\rho(\pi_{Ra},P^\star) \geq \rho(\pi_B,P^\star)$. Moreover, its performance loss $\Phi(\pi_{Ra})$ satisfies*

$$\Phi(\pi_{Ra}) \overset{\Delta}{=} \rho(\pi^\star,P^\star) - \rho(\pi_{Ra},P^\star) \leq \min\Big\{\frac{2\gamma R_{\max}}{(1-\gamma)^2}\Big(\|e_{\pi^\star}\|_{1,u^\star_{\pi^\star}} + \|e_{\pi_B}\|_{1,u^\star_{\pi_B}}\Big), \Phi(\pi_B)\Big\},$$

where $u^\star_{\pi^\star}$ is the state occupancy distribution of the optimal policy $\pi^\star$ in the true MDP $P^\star$, and $\Phi(\pi_B) = \rho(\pi^\star_{\xi^\star},P^\star) - \rho(\pi_B,P^\star)$ is the performance loss of the baseline policy.

Proof. To prove the safety of π_{Ra} and bound its performance loss, we need to upper and lower bound the difference between the performance of any policy π in the true MDP $P^\star$ and its performance in $\widetilde{\xi}$, i.e., $\rho(\pi, P^\star) - \rho(\pi, \widetilde{\xi})$. These upper and lower bounds are obtained by applying Lemma 10 with $P_1 = P^\star$, and $P_2 = \widetilde{\xi}$ as follows:

$$\rho(\pi, P^\star) - \rho(\pi, \widetilde{\xi}) \geq p_0^\mathsf{T}(\mathbf{I} - \gamma P_\pi^\star)^{-1}\left(r_\pi - \widehat{r}_\pi - \frac{\gamma R_{\max}}{1 - \gamma}e_\pi\right) \geq 0, \tag{14}$$

where the second inequality in (14) follows from the definition of the adjusted reward function $\widehat{r}$, and the fact that $(\mathbf{I} - \gamma P_\pi^\star)^{-1}$ is monotone and p_0 is non-negative. Similarly, the upper-bound may be written as

$$\rho(\pi, P^\star) - \rho(\pi, \widetilde{\xi}) \leq \frac{2\gamma R_{\max}}{1 - \gamma}p_0^\mathsf{T}(\mathbf{I} - \gamma P_\pi^\star)e_\pi = \frac{2\gamma R_{\max}}{(1 - \gamma)^2}\|e_\pi\|_{1, u_\pi^\star}, \tag{15}$$

where $u_\pi^\star = (1 - \gamma)p_0^\mathsf{T}(\mathbf{I} - \gamma P_\pi^\star)^{-1}$ is the state occupancy distribution of policy π in the true MDP $P^\star$.

To prove the safety of the returned policy π_{Ra}: Consider the two cases on Line 4 of Algorithm 3. When the condition is satisfied, we have

$$\rho(\pi_B, P^\star) \leq \max_{\xi \in \Xi}\rho(\pi_B, \xi) < \rho(\pi_0, \widetilde{\xi}) \leq \rho(\pi_0, P^\star),$$

where the last inequality comes from (14), and thus, the policy $\pi_{Ra} = \pi_0$ is *safe*. When the condition is violated, then π_{Ra} is simply π_B, which is safe by definition.

To derive a bound on the performance loss of the returned policy π_{Ra}: Consider also the two cases on Line 4 of Algorithm 3. When the condition is satisfied, using (14), we have

$$\Phi(\pi_{Ra}) \overset{\Delta}{=} \rho(\pi^\star, P^\star) - \rho(\pi_{Ra}, P^\star) = \rho(\pi^\star, P^\star) - \rho(\pi_0, P^\star) \leq \rho(\pi^\star, P^\star) - \rho(\pi_0, \widetilde{\xi}),$$

and when the condition is violated, we have

$$\Phi(\pi_{Ra}) \overset{\Delta}{=} \rho(\pi^\star, P^\star) - \rho(\pi_{Ra}, P^\star) = \rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star).$$

Since when the condition is satisfied on Line 4 of Algorithm 3, we have

$$\rho(\pi_0, \widetilde{\xi}) > \max_{\xi \in \Xi}\rho(\pi_B, \xi),$$

in both cases on Line 4 of Algorithm 3, we may write

$$\Phi(\pi_{Ra}) \leq \min\left\{\rho(\pi^\star, P^\star) - \rho(\pi_0, \widetilde{\xi}) + \max_{\xi \in \Xi}\rho(\pi_B, \xi) - \rho(\pi_B, P^\star), \overbrace{\rho(\pi^\star, P^\star) - \rho(\pi_B, P^\star)}^{\Phi(\pi_B)}\right\}.$$

The first term in the minimum may be written as

$$\rho(\pi^\star, P^\star) - \rho(\pi_0, \widetilde{\xi}) + \max_{\xi \in \Xi}\rho(\pi_B, \xi) - \rho(\pi_B, P^\star)$$

$$\overset{(a)}{\leq}\rho(\pi^\star, P^\star) - \rho(\pi^\star, \widetilde{\xi}) + \max_{\xi \in \Xi}\rho(\pi_B, \xi) - \rho(\pi_B, P^\star) \overset{(b)}{\leq} \frac{2\gamma R_{\max}}{(1 - \gamma)^2}\|e_{\pi^\star}\|_{1, u_{\pi^\star}^\star} + \frac{2\gamma R_{\max}}{(1 - \gamma)^2}\|e_{\pi_B}\|_{1, u_{\pi_B}^\star},$$

where (a) follows from π_0 being an optimal policy of RaMDP $\widetilde{\xi}$ and (b) is from (15) with $\pi = \pi^\star$ and $\pi = \pi_B$. □

Theorem 14 indicates that by this simple adjustment in the reward function of the simulator $\widehat{P}$, we may guarantee that our solution is *safe*. Moreover, it shows that the bound on the performance loss of π_{Ra} is actually tighter than that for the solution π_{sim} of the simulator, reported in Theorem 8.

While Algorithm 2 is more complex than Algorithm 3 (since solving a RMDP is more complicated than a standard MDP), Theorem 12 does not show any advantage for π_R over π_{RA}, neither in terms of safety nor in terms of the bound on its performance loss (compared to Theorem 14). On the other

hand, while Algorithm 3 guarantees to yield a safe policy more efficiently than Algorithm 2, from Proposition 13 one notices that Algorithm 3 may be overly conservative in many circumstances. This is because the adjustment of the reward function is based on the assumption that there exists a state with the maximum value of $R_{\max}/(1 - \gamma)$ and that this state is accessible from all other states with reward $R_{\max}$. Thus, we may conclude that Algorithm 2 returns a less conservative safe policy (compared to Algorithm 3), with extra computational cost.

The experimental results of Section 5 also show that the reward-adjusted solution of Algorithm 3 can be extremely conservative.

I Description of Experimental Domains

I.1 Grid Problem

We now describe the grid problem in more detail. The state space in the problem comes from a two-dimensional grid: $\mathcal{S} = \{s_{ij} \ : \ i \in \mathcal{I}, j \in \mathcal{J}\}$; i and j represent a column and row respectively. Columns represent states of an interaction with the website, and rows represent more complex customer states, such as overall satisfaction. The dimensions are $|\mathcal{I}| = 12$ and $\mathcal{J} = 3$.

There are 4 actions: a_L for left, a_R for right, a_U for up, and a_U for down. Rewards are independent of actions and depend only on states and only on the column: $x_{ij} = r_i$ where $r = [-1, 1, 2, 3, 2, 1, -1, -2, -3, 3, 4, 5]$. Actions left and right generally decrease and increase the column number; but can fail and in that case the transition is to a random column. The failure probability z_j depends on the row j, with specific failure probabilities: $z = [0.9, 0.2, 0.3]$. If a transition fails, then the next state is chosen according to a fixed distribution which is generated a priori from a Dirichlet distribution. The distribution for first and last row are the same, and the middle row is the average of the two.

Algorithm 4 describes how the transition from a state is computed. The initial state is s_{00}.

Algorithm 4: Transitions for state and action.

    Data: Current state s_{ij}, action a, distributions P_j
    Result: Next state s_{kl}
1 **if** *Random uniform from* $[0, 1] > z_j$ **then**
2   **if** $a = a_R$ **then**
3 | $k \leftarrow i + 1$;
4   **else if** $a = a_L$ **then**
5 | $k \leftarrow i - 1$;
6   **else**
7 | $k \leftarrow$ Random from P_j ;
8   **end**
9 $k \leftarrow \max\{0, \min\{k, |\mathcal{I}| - 1\}\}$;
10 **else**
11 | $k \leftarrow$ Random from P_j ;
12 **end**
13 **if** $a = a_U$ **then**
14 | $l \leftarrow j + 1$;
15 **else if** $a = a_D$ **then**
16 | $l \leftarrow j - 1$;
17 **else**
18 $e \leftarrow$ Random uniform from $[0, 1]$;
19   **if** $e \leq 0.35$ **then**
20 | $l \leftarrow j + 1$;
21   **else if** $e \leq 0.7$ **then**
22 | $l \leftarrow j - 1$;
23   **else**
24 | $l \leftarrow j$;
25   **end**
26 **end**
27 $l \leftarrow \max\{0, \min\{l, |\mathcal{J}| - 1\}\}$;
28 **return** s_{kl};

I.2 Energy Arbitrage

The energy arbitrage model is based on [Petrik and Wu, 2015] using a discount factor 0.9999. We summarize it here for ease of reference. Recall that even though the state and action spaces in this problem are continuous, we discretize them as described in Section 5.

The problem represents an energy arbitrage model with multiple finite *known* price levels and a stochastic evolution given a limited storage capacity. In particular, the storage is assumed to be an electrical battery that degrades when energy is stored or retrieved. Energy prices are governed by a Markov process with states Θ. There are two energy prices in each time step: $p^i : \Theta \to \mathbb{R}_+$ is the purchase (or input) price and $p^o : \Theta \to \mathbb{R}_+$ is the sell (or output) price. Energy prices θ vary between 0 and 10 and their evolution is governed by a martingale with a normal distribution around the mean.

We use s to denote the available battery capacity with s_0 denoting the initial capacity. The current state of charge is denotes as z or y and must satisfy that $0 \le z_t \le s_t$ at any time step t. The action is the amount of energy to charge or discharge, which is denoted by a. A positive a indicates that energy is purchased to charge the battery; a negative a indicates the sale of energy.

The battery storage degrades with use. The degradation is a function of the battery capacity when charged or discharged. We use a general model of battery degradation with a specific focus on Li-ion batteries. The degradation function $d(z, a) \in \mathbb{R}_+$ represent the battery capacity loss after starting at the state of charge $z \ge 0$ and charging (discharging if negative) by a with $-z \le a \le s_0$. This function indicates the loss of capacity, such that:

$$s_{t+1} = s_t - d(z_t, a_t)$$

The state set in the Markov decision problem is composed of (z, s, θ) where z is the state of charge, s is the battery capacity, and $\theta \in \Theta$ is the state of the price process. The available actions in a state (z, s, θ) are a such that $-z \le a \le s - z$. The transition is from (z_t, s_t, θ_t) to $(z_{t+1}, s_{t+1}, \theta_{t+1})$ given action a_t is:

$$z_{t+1} = z_t + a_t$$
$$s_{t+1} = s_t - d(z_t, a_t)$$

The probability of this transition is given by $P[\theta_{t+1}|\theta_t]$. The reward for this transition is:

$$r((z_t, s_t, \theta_t), a_t) = \begin{cases} -a_t \cdot p^i - c^d \cdot d(z_t, a_t) & \text{if } a_t \ge 0 \\ -a_t \cdot p^o - c^d \cdot d(z_t, a_t) & \text{if } a_t < 0 \end{cases}.$$

That is, the reward captures the monetary value of the transaction minus a penalty for degradation of the battery. Here, c^d represents the cost of a unit of lost battery capacity.

The Bellman optimality equations for this problem are:

$$\begin{aligned}
q_T(z, s, \theta) &= 0 \\
v_t(z, s, \theta_t) &= \min\{p_{\theta_t}^i [a]_+ + p_{\theta_t}^o [a]_- + \\
&\quad + c^d d(z, a) + \\
&\quad + q_t(z + a, s - d(z, a), \theta_t) : \\
&\quad : a \in [-z, s - z]\} \\
q_t(z, s, \theta_t) &= \lambda \cdot \mathrm{E}[v_{t+1}(z, s, \theta_{t+1})]
\end{aligned} \qquad (16)$$

where $[a]_+ = \max\{a, 0\}$ and $[a]_- = \min\{a, 0\}$ and the expectation is taken over $P(\theta_{t+1}|\theta_t)$.

Please see [Petrik and Wu, 2015] for more details, including the price transition matrix.