Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Robust Control of Markov Decision Processes with Uncertain Transition Matrices

Arnab Nilim, Laurent El Ghaoui,

To cite this article:
Arnab Nilim, Laurent El Ghaoui, (2005) Robust Control of Markov Decision Processes with Uncertain Transition Matrices. Operations Research 53(5):780-798. https://doi.org/10.1287/opre.1050.0216

OPERATIONS RESEARCH
Vol. 53, No. 5, September–October 2005, pp. 780–798
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5305 | 0780

DOI 10.1287/opre.1050.0216

Robust Control of Markov Decision Processes with Uncertain Transition Matrices

Arnab Nilim, Laurent El Ghaoui
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720
{nilim@eecs.berkeley.edu, elghaoui@eecs.berkeley.edu}

Optimal solutions to Markov decision problems may be very sensitive with respect to the state transition probabilities. In many practical problems, the estimation of these probabilities is far from accurate. Hence, estimation errors are limiting factors in applying Markov decision processes to real-world problems.

We consider a robust control problem for a finite-state, finite-action Markov decision process, where uncertainty on the transition matrices is described in terms of possibly nonconvex sets. We show that perfect duality holds for this problem, and that as a consequence, it can be solved with a variant of the classical dynamic programming algorithm, the "robust dynamic programming" algorithm. We show that a particular choice of the uncertainty sets, involving likelihood regions or entropy bounds, leads to both a statistically accurate representation of uncertainty, and a complexity of the robust recursion that is almost the same as that of the classical recursion. Hence, robustness can be added at practically no extra computing cost. We derive similar results for other uncertainty sets, including one with a finite number of possible values for the transition matrices.

We describe in a practical path planning example the benefits of using a robust strategy instead of the classical optimal strategy; even if the uncertainty level is only crudely guessed, the robust strategy yields a much better worst-case expected travel time.

Subject classifications: dynamic programming: Markov, finite state, game theory; programming: convex, uncertainty, robustness; statistics: estimation.
Area of review: Stochastic Models.
History: Received January 2003; revisions received January 2004, May 2004; accepted September 2004.

Notation

$P > 0$ or $P \geqslant 0$ refers to the strict or nonstrict componentwise inequality for matrices or vectors. For a vector $p > 0$, $\log p$ refers to the componentwise operation. The notation $\mathbf{1}$ refers to the vector of ones, with size determined from context. The probability simplex in $\mathbf{R}^n$ is denoted $\Delta_n = \{p \in \mathbf{R}^n_+ : p^T\mathbf{1} = 1\}$, while Θ_n is the set of $n \times n$ transition matrices (componentwise nonnegative matrices with rows summing to one). We use $\sigma_{\mathscr{P}}$ to denote the support function of a set $\mathscr{P} \subseteq \mathbf{R}^n$, with for $v \in \mathbf{R}^n$, $\sigma_{\mathscr{P}}(v) := \sup\{p^T v : p \in \mathscr{P}\}$.

1. Introduction

Finite-state and finite-action Markov decision processes (MDPs) capture several attractive features that are important in decision making under uncertainty: they handle risk in sequential decision making via a state transition probability matrix, while taking into account the possibility of information gathering and using this information to apply recourse during the multistage decision process (Putterman 1994, Berstsekas and Tsitsiklis 1996, Mine and Osaki 1970, Feinberg and Shwartz 2002).

This paper addresses the issue of uncertainty at a higher level: We consider a Markov decision problem in which the transition probabilities themselves are uncertain, and seek a robust decision for it. Our work is motivated by the fact that in many practical problems, the transition matrices have to be estimated from data, and this may be a difficult task; see, for example, Kalyanasundaram et al. (2001), Feinberg and Shwartz (2002), Abbad and Filar (1992), and Abbad et al. (1992). It turns out that estimation errors may have a huge impact on the solution, which is often quite sensitive to changes in the transition probabilities. We will provide an example of this phenomenon in §8.

A number of authors have addressed the issue of uncertainty in the transition matrices of an MDP. A Bayesian approach such as described by Shapiro and Kleywegt (2002) requires a perfect knowledge of the whole prior distribution on the transition matrix, making it difficult to apply in practice. Other authors have considered the transition matrix to lie in a given set, most typically a polytope (Satia and Lave 1973, White and Eldeib 1994, Givan et al. 1997). Although our approach allows one to describe the uncertainty on the transition matrix by a polytope, we will argue *against* choosing such a model for the uncertainty. First, a general polytope is often not a tractable way to address the robustness problem, as it incurs a significant additional computational effort to handle uncertainty. As we will show, an exception is when the uncertainty is described by an interval matrix, intersected

by the constraint that probabilities sum to one, as in Givan et al. (1997) and Bagnell et al. (2001); or, when the polytope is described by its vertices. Perhaps more importantly, polytopic models, especially interval matrices, may be very poor representations of statistical uncertainty and lead to very conservative robust policies (Nilim and El Ghaoui 2002). In Bagnell et al. (2001), authors consider a problem dual to ours, and give without proof the "robust value iteration," which we derive here. Like us, they consider relative entropy as a way to measure uncertainties in the transition matrices; however, they do not propose any specific algorithm to solve the corresponding "inner problem," which has to be solved at each step of the robust value iteration. They only provide a general statement according to which the cost of solving the inner problem is polynomial in problem size, provided the uncertainty on the transition matrices is described by convex sets. In Iyengar (2003), the author discusses a problem similar to ours, introducing two versions of uncertainty (static and dynamic), and provides an independent proof of the robust value iteration in the case of compact uncertainty sets.

2. Problem Setup

2.1. Nominal Problem

We consider a finite-horizon MDP with finite decision horizon $T = \{0, 1, 2, \ldots, N-1\}$. At each stage, the system occupies a state $i \in \mathscr{X}$, where $n = |\mathscr{X}|$ is finite, and a decision maker is allowed to choose an action a deterministically from a finite set of allowable actions $\mathscr{A} = \{a_1, \ldots, a_m\}$ (for notational simplicity we assume that $\mathscr{A}$ is not state dependent). The system starts in a given initial state i_0. The states make Markov transitions according to a collection of (possibly time dependent) transition matrices $\tau := (P_t^a)_{a\in\mathscr{A},\, t\in T}$, where for every $a \in \mathscr{A}$, $t \in T$, the $n \times n$ transition matrix P_t^a contains the probabilities of transition under action a at stage t. We denote by $\pi = (\mathbf{a}_0, \ldots, \mathbf{a}_{N-1})$ a generic controller policy, where $\mathbf{a}_t(i)$ denotes the controller action when the system is in state $i \in \mathscr{X}$ at time $t \in T$. Let $\Pi = \mathscr{A}^{nN}$ be the corresponding strategy space. Define by $c_t(i, a)$ the cost corresponding to state $i \in \mathscr{X}$ and action $a \in \mathscr{A}$ at time $t \in T$, and by c_N the cost function at the terminal stage. We assume that $c_t(i, a)$ is nonnegative and finite for every $i \in \mathscr{X}$ and $a \in \mathscr{A}$.

For a given set of transition matrices τ, we define the finite-horizon *nominal* problem by

$$\phi_N(\Pi, \tau) := \min_{\pi\in\Pi} C_N(\pi, \tau), \tag{1}$$

where $C_N(\pi, \tau)$ denotes the *expected total cost* under controller policy π and transitions τ:

$$C_N(\pi, \tau) := \mathbf{E}\left(\sum_{t=0}^{N-1} c_t(i_t, \mathbf{a}_t(i)) + c_N(i_N)\right). \tag{2}$$

A special case of interest is when the expected total cost function bears the form (2), where the terminal cost is zero, and $c_t(i, a) = \nu^t c(i, a)$, with $c(i, a)$ now a constant cost function, which we assume nonnegative and finite everywhere, and $\nu \in (0, 1)$ is a discount factor. We refer to this cost function as the discounted cost function, and denote by $C_\infty(\pi, \tau)$ the limit of the discounted cost (2) as $N \to \infty$.

When the transition matrices are exactly known, the corresponding nominal problem can be solved via a dynamic programming algorithm, which has total complexity of mn^2N flops in the finite-horizon case. In the infinite-horizon case with a discounted cost function, the cost of computing an ϵ-suboptimal policy via the Bellman recursion is $O(mn^2 \log(1/\epsilon))$; see Putterman (1994) for more details.

2.2. Robust Control Problems

First, we consider the finite-horizon case, and assume that when for each action a and time t, the corresponding transition matrix P_t^a is only known to lie in some given subset $\mathscr{P}^a$ of Θ_n. Loosely speaking, we can think of the sets $\mathscr{P}^a$ as *sets of confidence* for the transition matrices. We further assume that the sets $\mathscr{P}^a$ satisfy:

Rectangular Uncertainty Property. For every $a \in \mathscr{A}$, $\mathscr{P}^a$ has the form $\mathscr{P}^a = \mathscr{P}_1^a \times \cdots \times \mathscr{P}_n^a$, where $\mathscr{P}_i^a$s are given subsets of the probability simplex in $\mathbf{R}^n$ that describe the uncertainty on the ith row of P^a (that is, on the state distribution given action a).

Note that our uncertainty model does not allow for correlations between the uncertainties affecting the P^as across different actions a, nor between different rows of each matrix.

Two models for transition matrix uncertainty are possible, leading to two possible forms of finite-horizon robust control problems. In a first model, referred to as the *stationary uncertainty model*, the transition matrices are chosen by nature depending on the controller policy once and for all, and remain fixed thereafter. In a second model, which we refer to as the *time-varying uncertainty model*, the transition matrices can vary arbitrarily with time, within their prescribed bounds. Each problem leads to a game between the controller and nature, where the controller seeks to minimize the maximum expected cost, with nature being the maximizing player.

Let us define our two problems more formally. A *policy of nature* refers to a specific collection of time-dependent transition matrices $\tau = (P_t^a)_{a\in\mathscr{A},\, t\in T}$ chosen by nature, and the set of admissible policies of nature is $\mathscr{T} := (\bigotimes_{a\in\mathscr{A}} \mathscr{P}^a)^N$, where $\bigotimes$ denotes direct product. Denote by $\mathscr{T}_s$ the set of stationary admissible policies of nature:

$$\mathscr{T}_s = \left\{\tau = (P_t^a)_{a\in\mathscr{A},\, t\in T} \in \mathscr{T}: \right.$$
$$\left. P_t^a = P_s^a \text{ for every } t, s \in T,\ a \in \mathscr{A}\right\}.$$

The stationary uncertainty model leads to the problem

$$\phi_N(\Pi, \mathscr{T}_s) := \min_{\pi\in\Pi} \max_{\tau\in\mathscr{T}_s} C_N(\pi, \tau). \tag{3}$$

In contrast, the time-varying uncertainty model leads to a relaxed version of the above:

$$\phi_N(\Pi, \mathcal{T}_s) \leqslant \phi_N(\Pi, \mathcal{T}) := \min_{\pi\in\Pi} \max_{\tau\in\mathcal{T}} C_N(\pi, \tau). \tag{4}$$

The first model is attractive for statistical reasons, as it is much easier to develop statistically accurate sets of confidence when the underlying process is time invariant. Unfortunately, the resulting game (3) seems to be hard to solve. The second model is attractive as one can solve the corresponding game (4) using a variant of the dynamic programming algorithm seen later, but we are left with a difficult task, that of estimating a meaningful set of confidence for the time-varying matrices P_t^a. In this paper, we will use the first model of uncertainty to derive statistically meaningful sets of confidence for the transition matrices, based on likelihood or entropy bounds. Then, instead of solving the corresponding difficult control problem (3), we use an approximation that is common in robust control, and solve the time-varying upper bound (4), using the uncertainty sets $\mathcal{P}^a$ derived from a stationarity assumption about the transition matrices.

We will also consider a variant of the finite-horizon time-varying problem (4), where controller and nature play alternatively, leading to a sequential game

$$\phi_N^{\text{seq}}(\Pi, \mathcal{Q}) := \min_{\mathbf{a}_0} \max_{\tau_0\in\mathcal{Q}} \min_{\mathbf{a}_1} \max_{\tau_1\in\mathcal{Q}} \cdots \min_{\mathbf{a}_{N-1}} \max_{\tau_{N-1}\in\mathcal{Q}} C_N(\pi, \tau), \tag{5}$$

where the notation $\tau_t = (P_t^a)_{a\in\mathcal{A}}$ denotes the collection of transition matrices at a given time $t \in T$, and $\mathcal{Q} := \bigotimes_{a\in\mathcal{A}} \mathcal{P}^a$ is the corresponding uncertainty set from which nature is allowed to choose the transition matrices at every stage.

Finally, we will consider an infinite-horizon robust control problem, with the discounted cost function referred to above, and where we restrict control and nature policies to be stationary:

$$\phi_\infty(\Pi_s, \mathcal{T}_s) := \min_{\pi\in\Pi_s} \max_{\tau\in\mathcal{T}_s} C_\infty(\pi, \tau), \tag{6}$$

where Π_s denotes the space of stationary control policies. We define $\phi_\infty(\Pi, \mathcal{T})$, $\phi_\infty(\Pi, \mathcal{T}_s)$, and $\phi_\infty(\Pi_s, \mathcal{T})$ accordingly.

In the sequel, for a given control policy $\pi \in \Pi$ and subset $\mathcal{S} \subseteq \mathcal{T}$, the notation $\phi_N(\pi, \mathcal{S}) := \max_{\tau\in\mathcal{S}} C_N(\pi, \tau)$ denotes the worst-case expected total cost for the finite-horizon problem, and $\phi_\infty(\pi, \mathcal{S})$ is defined likewise.

2.3. Main Results and Outline

Our main contributions are as follows. First, we derive a recursion, the "robust dynamic programming" algorithm, which solves the finite-horizon robust control problem (4). We provide a simple proof of the optimality of the recursion, where the main ingredient is to show that perfect duality holds in the game (4). (For completeness, another proof, which requires a theorem from stochastic game theory, is given in Appendix A.) As a corollary of this result, we obtain that the sequential game (5) is equivalent to its nonsequential counterpart (4). Second, we derive similar results for the infinite-horizon problem with discounted cost function, (6). Moreover, we obtain that if we consider a finite-horizon problem with a discounted cost function, then the gap between the optimal value of the stationary uncertainty problem (3) and that of its time-varying counterpart (4) goes to zero as the horizon length goes to infinity, at a rate determined by the discount factor. Finally, we identify several classes of uncertainty models, which result in an algorithm that is *both* statistically accurate and numerically tractable. We derive precise complexity results that imply that, with the proposed approach, robustness can be handled at practically no extra computing cost.

Our paper is organized as follows. Section 3 deals with the finite-horizon problem, including the "robust dynamic programming" theorem (Theorem 1) and its proof, as well as a detailed complexity analysis. Section 4 provides similar results for the infinite-horizon case. Sections 5 and 6 are devoted to specific uncertainty models, involving likelihood regions or entropy bounds, while §7 deals with finite scenario, ellipsoidal, and interval matrix models. We describe numerical results in the context of aircraft routing in §8. Section 9 contains concluding remarks.

3. Finite-Horizon Problem

We consider the finite-horizon robust control problem defined in §2.2. For a given state $i \in \mathcal{X}$, action $a \in \mathcal{A}$, and $P^a \in \mathcal{P}^a$, we denote by p_i^a the next-state distribution drawn from P^a corresponding to state $i \in \mathcal{X}$; thus p_i^a is the ith row of matrix P^a. We define $\mathcal{P}_i^a$ as the projection of the set $\mathcal{P}^a$ onto the set of p_i^a-variables; by the rectangular uncertainty property, $\mathcal{P}^a$ is the direct product of these sets. By assumption, $\mathcal{P}_i^a$s are included in the probability simplex of $\mathbf{R}^n$, Δ_n; no other property is assumed.

3.1. Robust Dynamic Programming

We provide below a self-contained proof of the following theorem, based on linear programming duality. For completeness, we provide an alternate proof in Appendix A, based on a stochastic game formulation. Yet another proof of the robust Bellman recursion (7), (8) is also given by Iyengar (2003), via an appropriately defined robust value function and exploiting a certain "rectangularity property" (Epstein and Schneider 2002), which is different from the rectangular uncertainty property defined in §2.2.

THEOREM 1 (ROBUST DYNAMIC PROGRAMMING). *For the robust control problem* (4), *perfect duality holds*:

$$\begin{aligned}\phi_N(\Pi, \mathcal{T}) &= \min_{\pi\in\Pi} \max_{\tau\in\mathcal{T}} C_N(\pi, \tau) \\ &= \max_{\tau\in\mathcal{T}} \min_{\pi\in\Pi} C_N(\pi, \tau) := \psi_N(\Pi, \mathcal{T}).\end{aligned}$$

The problem can be solved via the recursion

$$v_t(i) = \min_{a\in\mathscr{A}}\big(c_t(i,a) + \sigma_{\mathscr{P}_i^a}(v_{t+1})\big), \quad i\in\mathscr{X},\ t\in T, \tag{7}$$

where $\sigma_{\mathscr{P}}(v) := \sup\{p^T v\colon p\in\mathscr{P}\}$ *denotes the support function of a set* $\mathscr{P}$*, and* $v_t(i)$ *is the worst-case optimal value function in state* i *at stage* t*. A corresponding optimal control policy* $\pi^* = (\mathbf{a}_0^*, \ldots, \mathbf{a}_{N-1}^*)$ *is obtained by setting*

$$\mathbf{a}_t^*(i) \in \arg\min_{a\in\mathscr{A}}\big\{c_t(i,a) + \sigma_{\mathscr{P}_i^a}(v_{t+1})\big\}, \quad i\in\mathscr{X}, \tag{8}$$

and a corresponding worst-case nature policy is obtained by choosing the i*th row of the transition matrix* P_t^a *as*

$$p_i^a(t) \in \arg\max_p\{p^T v_{t+1}\colon p\in\mathscr{P}_i^a\}, \quad i\in\mathscr{X},\ a\in\mathscr{A},\ t\in T. \tag{9}$$

The effect of uncertainty on a given strategy $\pi = (\mathbf{a}_0, \ldots, \mathbf{a}_N)$ *can be evaluated by the following recursion*:

$$v_t^\pi(i) = c_t(i, \mathbf{a}_t(i)) + \sigma_{\mathscr{P}_i^{\mathbf{a}_t(i)}}(v_{t+1}^\pi), \quad i\in\mathscr{X}, \tag{10}$$

which provides the worst-case value function v^π *for the strategy* π*.*

Proof. We begin with a simple technical lemma.

Lemma 1. *For given* $v_N\in\mathbf{R}^n$*, consider the problem*

$$\eta := \max_{v_0,\ldots,v_{N-1}} q^T v_0\colon\ v_t \leqslant g_t(v_{t+1}), \quad t\in T,\ i\in\mathscr{X}, \tag{11}$$

where inequalities are understood componentwise, $q\in\mathbf{R}_+^n$*, and the functions* $g_t\colon \mathbf{R}^n\to\mathbf{R}^n$ *are given. If the functions* g_t *are componentwise nondecreasing for every* $t\in T$*, meaning that* $g_t(u)\leqslant g_t(v)$ *for every* $u, v\in\mathbf{R}^n$ *with* $u\leqslant v$*, then the optimal variables can be computed via the recursion*

$$v_t = g_t(v_{t+1}), \quad t\in T, \tag{12}$$

and the optimal value is $\eta = q^T(g_1\circ\cdots\circ g_N)(v_N)$*.*

To prove Lemma 1, we note that recursion (12) yields $v_0 = v_0^* := (g_1\circ\cdots\circ g_N)(v_N)$. In addition, this recursion provides a feasible point for the problem, hence $\eta\geqslant q^T v_0^*$. Because $q\geqslant 0$, and each g_t is componentwise nondecreasing, we also have $\eta\leqslant q^T v_0^*$, which shows that the recursion provides the optimal value of problem (11). This proves the lemma.

We proceed with a well-known linear programming representation of the nominal problem (1) (Putterman 1994):

$$\begin{aligned}\phi_N(\Pi,\tau) &:= \max_{v_0,\ldots,v_{N-1}} q^T v_0\colon\ v_t(i)\leqslant c_t(i,a) + \sum_j P_t^a(i,j)v_{t+1}(j),\\ &\qquad a\in\mathscr{A},\ i\in\mathscr{X},\ t\in T,\end{aligned} \tag{13}$$

where q is a componentwise nonnegative vector, precisely $q(i) = 0$ if $i\neq i_0$, $q(i_0) = 1$, where i_0 is the initial state. In the above, we have denoted by $\tau := (P_t^a)_{a\in\mathscr{A},\,t\in T}$ the (given) collection of time-varying transition matrices. Likewise, the expected cost for a *given* controller policy $\pi = (\mathbf{a}_t)_{t\in T}$ is given by the linear program

$$\begin{aligned}\phi_N(\pi,\tau) &:= \max_{v_0,\ldots,v_{N-1}} q^T v_0\colon\ v_t(i)\leqslant c_t(i,\mathbf{a}_t(i))\\ &\qquad + \sum_j P_t^{\mathbf{a}_t(i)}(i,j)v_{t+1}(j), \quad i\in\mathscr{X},\ t\in T.\end{aligned} \tag{14}$$

By weak duality, $\phi_N(\Pi,\mathscr{T})\geqslant\psi_N(\Pi,\mathscr{T})$, where $\psi_N(\Pi,\mathscr{T})$ is defined in the theorem. Let us prove that perfect duality holds, that is, $\phi_N(\Pi,\mathscr{T}) = \psi_N(\Pi,\mathscr{T})$. The lower bound $\psi_N(\Pi,\mathscr{T})$ can be expressed as the optimal value of the following nonlinear problem (in variables v, τ):

$$\begin{aligned}&\psi_N(\Pi,\mathscr{T})\\ &\quad := \max_{\tau\in\mathscr{T},\,v_0,\ldots,v_{N-1}} q^T v_0\colon\ v_t(i)\leqslant c_t(i,a) + \sum_j P_t^a(i,j)v_{t+1}(j),\\ &\qquad\qquad a\in\mathscr{A},\ i\in\mathscr{X},\ t\in T.\end{aligned} \tag{15}$$

The difference between the nominal problem (13) and (15) is simply that the matrices P_t^a are fixed in (13), while they are variables in problem (15).

Denote by $\phi_N(\pi,\mathscr{T}) = \max_{\tau\in\mathscr{T}} C_N(\pi,\tau)$ the worst-case expected total cost for a given policy π. This value is obtained by letting the matrices $P_t^{\mathbf{a}_t(i)}$ become variables in (14), which results in

$$\begin{aligned}\phi_N(\pi,\mathscr{T}) &:= \max_{\tau\in\mathscr{T},\,v_0,\ldots,v_{N-1}} q^T v_0\colon\ v_t(i)\leqslant c_t(i,\mathbf{a}_t(i))\\ &\qquad + \sum_j P_t^{\mathbf{a}_t(i)}(i,j)v_{t+1}(j), \quad i\in\mathscr{X},\ t\in T.\end{aligned} \tag{16}$$

Due to the rectangular uncertainty property $\mathscr{P}^a = \mathscr{P}_1^a\times\cdots\times\mathscr{P}_n^a$, the problem of computing $\psi_N(\Pi,\mathscr{T})$, and that of computing $\phi_N(\pi,\mathscr{T})$ for a given policy π, can both be represented as problem (11) of Lemma 1, where we define the functions g_t, $t\in T$, by their components, as follows for problem (15):

$$(g_t(v))_i := \min_{a\in\mathscr{A}}\big(c_t(i,a) + \sigma_{\mathscr{P}_i^a}(v)\big), \quad i\in\mathscr{X},$$

and as follows for problem (16):

$$(g_t(v))_i := c_t(i,\mathbf{a}_t(i)) + \sigma_{\mathscr{P}_i^{\mathbf{a}_t(i)}}(v), \quad i\in\mathscr{X}.$$

Because the sets $\mathscr{P}_i^a$ are all included in Δ^n, the above functions are componentwise nondecreasing, and Lemma 1 applies. This shows that problems (15) and (16) can be solved by the recursions (7) and (10), respectively, as given in Theorem 1.

Recursion (7) provides a policy $\pi^* = (\mathbf{a}_0^*, \ldots, \mathbf{a}_{N-1}^*)$, via expression (8) as given in the theorem. We can express the recursion exactly as in (10), with $\mathbf{a}_t$ replaced with $\mathbf{a}_t^*$, $t\in T$.

This shows that $\psi_N(\Pi, \mathcal{T}) = \phi_N(\pi^*, \mathcal{T})$. Because π^* is an admissible (that is, deterministic) policy, we necessarily have $\phi_N(\pi^*, \mathcal{T}) \geqslant \phi_N(\Pi, \mathcal{T})$. This shows that perfect duality holds: $\phi_N(\Pi, \mathcal{T}) = \phi_N(\pi^*, \mathcal{T}) = \psi_N(\Pi, \mathcal{T})$, and that the policy π^* provided by expression (8) is optimal for the robust control problem (4).

Finally, the expression for the optimal worst-case policy of nature given in (9) is obtained by noting that it corresponds to the solution of problem (15) when π is set to the optimal control policy. This ends our proof. □

Note that our proof does not require convexity of the uncertainty sets $\mathcal{P}_i^a$; we only used the fact that these sets are entirely included in the probability simplex of $\mathbf{R}^n$.

We are ready to examine the sequential game (5).

COROLLARY 1. *The sequential game* (5) *is equivalent to the game* (4):

$$\phi_N^{\text{seq}}(\Pi, \mathcal{Q}) = \phi_N(\Pi, \mathcal{T}),$$

and the optimal strategies for $\phi_N(\Pi, \mathcal{T})$ *given in Theorem* 1 *are optimal for* $\phi_N^{\text{seq}}(\Pi, \mathcal{Q})$ *as well.*

PROOF. A repeated application of weak duality shows the lower bound $\phi_N^{\text{seq}}(\Pi, \mathcal{Q}) \leqslant \phi_N(\Pi, \mathcal{T})$ (this is simply a consequence of the fact that the sequential game gives less power to nature). Because the optimal worst-case nature strategy defined in Theorem 1 is feasible for problem (5), the result follows. □

3.2. Solving the Inner Problem

Each step of the robust dynamic programming algorithm involves the solution of an optimization problem, referred to as the "inner problem," of the form

$$\sigma_{\mathcal{P}}(v) = \max_{p \in \mathcal{P}} v^T p, \tag{17}$$

where the variable p corresponds to a particular row of a specific transition matrix, $\mathcal{P} = \mathcal{P}_i^a$ is the set that describes the uncertainty on this row, and v contains the elements of the value function at some given stage. Note that we can safely replace $\mathcal{P}$ in (17) by its convex hull, so that convexity of the sets $\mathcal{P}_i^a$ is not required; the algorithm only requires the knowledge of their convex hulls. The shape of the convex hulls $\mathbf{conv}(\mathcal{P}_i^a)$ for each $i \in \mathcal{X}$ and $a \in \mathcal{A}$ is a key component in the computational complexity of the robust dynamic programming algorithm.

Beyond numerical tractability, an additional criteria for the choice of a specific uncertainty model is that the sets $\mathcal{P}^a$ should represent accurate (nonconservative) descriptions of the statistical uncertainty on the transition matrices. Perhaps surprisingly, there are statistical models of uncertainty that are good on both counts; specific examples of such models are described in §§5 and 6. Precisely, the uncertainty models considered in §§5 and 6 all result in inner problems (17) that can be solved in worst-case time of $O(n \log(v_{\max}/\delta))$ via a simple bisection algorithm, where n is the size of the state space, $v_{\max}$ is a global upper bound on the value function, and $\delta > 0$ specifies the accuracy at which the optimal value of the inner problem (17) is computed. We defer the proof of this complexity result to the appropriate sections. The bisection algorithm can be interpreted as a function $\widehat{\sigma}_{\mathcal{P}}$ such that for every $v \in \mathbf{R}^n$, there exists $\delta_{\mathcal{P}}(v)$ such that

$$\widehat{\sigma}_{\mathcal{P}}(v) = \sigma_{\mathcal{P}}(v) + \delta_{\mathcal{P}}(v), \quad 0 \leqslant \delta_{\mathcal{P}}(v) \leqslant \delta. \tag{18}$$

3.3. Complexity Analysis

In this section, we discuss the complexity of computing an ϵ-suboptimal policy $\widehat{\pi}$, which is a policy such that the worst-case expected total cost under policy $\widehat{\pi}$, namely $\phi_N(\widehat{\pi}, \mathcal{T}) = \max_{\tau \in \mathcal{T}} C_N(\widehat{\pi}, \tau)$, satisfies $\phi_N(\widehat{\pi}, \mathcal{T}) - \epsilon \leqslant \phi_N(\Pi, \mathcal{T}) \leqslant \phi_N(\widehat{\pi}, \mathcal{T})$. Here, $\epsilon > 0$ is given. We assume that we use the specific uncertainty models considered in §§5 and 6, and that we solve the resulting inner problem with the bisection algorithm with an accuracy $\delta := \epsilon/N$.

THEOREM 2. *For the finite-horizon problem, if we solve the inner problem* (17) *with the bisection algorithm accuracy parameter* $\delta := \epsilon/N$, *our algorithm will guarantee an* ϵ*-suboptimal policy, with an additional computational cost of* $\log(N/\epsilon)$ *with respect to the classical dynamic programming algorithm.*

PROOF. When we apply the bisection algorithm within the robust dynamic programming algorithm given in §3.1, we generate vectors $\hat{v}_t$ by recursion (7), with $\sigma_{\mathcal{P}_i^a}$ replaced by $\widehat{\sigma}_{\mathcal{P}_i^a}$, as defined by (18). The corresponding Equation (8) yields a policy $\widehat{\pi}$. We can express the recursion that provides $\hat{v}$ as

$$\hat{v}_t(i) = \min_{a \in \mathcal{A}} \big(c_t(i, a) + \delta_t(i, a) + \sigma_{\mathcal{P}_i^a}(\hat{v}_{t+1}) \big), \quad i \in \mathcal{X},\ t \in T,$$

where $\delta_t(i, a) := \delta_{\mathcal{P}_i^a}(v_{t+1})$. The policy $\widehat{\pi}$ is obtained by looking at a minimizing index in the above. Thus, $\widehat{\pi}$ is optimal for the robust control problem (4), but with a modified cost function: $\hat{c}_t(i, a) = c_t(i, a) + \delta_t(i, a)$. The bounds $0 \leqslant \delta_t(i, a) \leqslant \epsilon/N$ then imply that the corresponding expected total cost function $\widehat{C}_N$ satisfies $C_N(\pi, \tau) \in [\widehat{C}_N(\pi, \tau) - \epsilon, \widehat{C}_N(\pi, \tau)]$ for every $\pi \in \Pi$ and $\tau \in \mathcal{T}$. Maximizing over τ for $\pi = \widehat{\pi}$ yields $\phi_N(\widehat{\pi}, \mathcal{T}) \in [\widehat{\phi} - \epsilon, \widehat{\phi}]$, where $\widehat{\phi} := \min_{\pi \in \Pi} \max_{\tau \in \mathcal{T}} \widehat{C}_N(\pi, \tau)$ is the optimal value of the modified control problem, and $\phi_N(\widehat{\pi}, \mathcal{T})$ is the worst-case expected total cost under policy $\widehat{\pi}$ for the original problem. Likewise, minimizing over π the maximum over τ yields $\phi_N(\Pi, \mathcal{T}) \in [\widehat{\phi} - \epsilon, \widehat{\phi}]$. Because $\phi_N(\Pi, \mathcal{T}) \leqslant \phi_N(\widehat{\pi}, \mathcal{T})$ because $\widehat{\pi}$ is deterministic, we conclude that $\phi_N(\Pi, \mathcal{T}) \in [\phi_N(\widehat{\pi}, \mathcal{T}) - \epsilon, \phi_N(\widehat{\pi}, \mathcal{T})]$.

We obtain that, to compute a suboptimal policy $\widehat{\pi}$ that achieves the exact optimum with prescribed accuracy ϵ, the number of flops required by the algorithm is $O(mn^2 N \log(v_{\max} N/\epsilon))$. The bound $v_{\max} \leqslant N C_{\max}$, with $C_{\max} = \max_{i \in \mathcal{X},\, a \in \mathcal{A},\, t \in T} c_t(i, a)$, then leads to the complexity

bound of $O(mn^2N\log(N/\epsilon))$, which means that robustness is obtained at a relative increase of computational cost of only $\log(N/\epsilon)$ with respect to the classical dynamic programming algorithm, which is small for moderate values of N. □

If N is very large, we can turn instead to the infinite-horizon problem examined in §4, and similar complexity results hold.

3.4. Algorithm

Our analysis yields an algorithm to compute an ϵ-suboptimal policy π^ϵ for problem (4) using the uncertainty models described in §§5 and 6. The algorithm has complexity $O(mn^2N\log(N/\epsilon))$.

Robust Finite-Horizon Dynamic Programming Algorithm

Step 1. Set $\epsilon > 0$. Initialize the value function to its terminal value $\hat{v}_N = c_N$.

Step 2. Repeat until $t = 0$:

(a) For every state $i \in \mathscr{X}$ and action $a \in \mathscr{A}$, compute, using the bisection algorithm described in SS5 or 6, a value $\hat{\sigma}_i^a$ such that

$$\hat{\sigma}_i^a - \epsilon/N \leqslant \sigma_{\mathscr{P}_i^a}(\hat{v}_t) \leqslant \hat{\sigma}_i^a.$$

(b) Update the value function by $\hat{v}_{t-1}(i) = \min_{a\in\mathscr{A}}(c_{t-1}(i,a) + \hat{\sigma}_i^a)$, $i \in \mathscr{X}$.

(c) Replace t by $t-1$ and go to Step 2.

Step 3. For every $i \in \mathscr{X}$ and $t \in T$, set $\pi^\epsilon = (\mathbf{a}_0^\epsilon, \ldots, \mathbf{a}_{N-1}^\epsilon)$, where

$$\mathbf{a}_t^\epsilon(i) = \arg\max_{a\in\mathscr{A}}\{c_{t-1}(i,a) + \hat{\sigma}_i^a\}, \quad i \in \mathscr{X},\ a \in \mathscr{A}.$$

4. Infinite-Horizon Problem

In this section, we address a the infinite-horizon robust control problem, with a discounted cost function of the form (2), where the terminal cost is zero, and $c_t(i,a) = \nu^t c(i,a)$, where $c(i,a)$ is now a constant cost function, which we assume nonnegative and finite everywhere, and $\nu \in (0,1)$ is a discount factor.

4.1. Robust Bellman Recursion

We begin with the infinite-horizon problem involving stationary control and nature policies defined in (6). In Bagnell et al. (2001), the authors consider the problem of computing the dual quantity $\psi_\infty(\Pi, \mathscr{T})$ defined below, and stated without proof that it can be computed by the recursion given in the theorem. The robust Bellman recursion for the infinite-horizon case (19, 20) is also proved independently in Iyengar (2003).

THEOREM 3 (ROBUST BELLMAN RECURSION). *For the infinite-horizon robust control problem* (6) *with stationary uncertainty on the transition matrices, stationary control policies, and a discounted cost function with discount factor* $\nu \in [0,1)$, *perfect duality holds*:

$$\phi_\infty(\Pi_s, \mathscr{T}_s) = \max_{\tau\in\mathscr{T}_s}\min_{\pi\in\Pi_s} C_\infty(\pi,\tau) := \psi_\infty(\Pi_s, \mathscr{T}_s).$$

The optimal value is given by $\phi_\infty(\Pi_s, \mathscr{T}_s) = v(i_0)$, *where* i_0 *is the initial state, and where the value function* v *satisfies is the optimality conditions*

$$v(i) = \min_{a\in\mathscr{A}}\big(c(i,a) + \nu\sigma_{\mathscr{P}_i^a}(v)\big), \quad i \in \mathscr{X}. \tag{19}$$

The value function is the unique limit value of the convergent vector sequence defined by

$$v_{k+1}(i) = \min_{a\in\mathscr{A}}\big(c(i,a) + \nu\sigma_{\mathscr{P}_i^a}(v_k)\big), \quad i \in \mathscr{X}, \quad k = 1, 2, \ldots. \tag{20}$$

A stationary, optimal control policy $\pi = (\mathbf{a}^*, \mathbf{a}^*, \ldots)$ *is obtained as*

$$\mathbf{a}^*(i) \in \arg\min_{a\in\mathscr{A}}\big\{c(i,a) + \nu\sigma_{\mathscr{P}_i^a}(v)\big\}, \quad i \in \mathscr{X}, \tag{21}$$

and a stationary optimal nature policy is obtained by choosing the i*th row of the transition matrix* P^a *as*

$$p_i^a \in \arg\max_p\{p^T v:\ p \in \mathscr{P}_i^a\}, \quad i \in \mathscr{X},\ a \in \mathscr{A}. \tag{22}$$

The effect of uncertainty on a given stationary strategy $\pi = (\mathbf{a}, \mathbf{a}, \ldots)$ *can be evaluated by the following equation*:

$$v^\pi(i) = c(i, \mathbf{a}(i)) + \nu\sigma_{\mathscr{P}_i^{\mathbf{a}(i)}}(v^\pi), \quad i \in \mathscr{X}, \tag{23}$$

which provides the worst-case value function for the strategy π.

PROOF. The proof follows identical lines as that of Theorem 1. As before, we begin with a simple technical lemma, which we state without proof.

LEMMA 2. *For a given vector* $q \in \mathbf{R}_+^n$ *and function* g: $\mathbf{R}^n \to \mathbf{R}^n$, *consider the problem*

$$\max_v q^T v:\ v \leqslant g(v), \tag{24}$$

where inequalities are understood componentwise. If the above problem is feasible, and g *is monotone nondecreasing and contractive, then there is a unique optimizer* v_∞, *which is the unique solution to the fixed-point equation* $v = g(v)$.

We then express the nominal problem (without uncertainty on the transition matrices) with the linear program

$$\max_v q^T v:\ v(i) \leqslant c(i,a) + \nu\sum_j P^a(i,j)v(j), \quad a \in \mathscr{A},\ i \in \mathscr{X}, \tag{25}$$

where q is a componentwise nonnegative vector, precisely $q(i) = 0$ if $i \neq i_0$, $q(i_0) = 1$, where i_0 is the initial state. Likewise, the expected cost for a *given* stationary controller policy $\pi = (\mathbf{a}, \mathbf{a}, \ldots)$ is given by the linear program

$$\max_v q^T v:\ v(i) \leqslant c(i, \mathbf{a}(i)) + \nu \sum_j P^{\mathbf{a}(i)}(i, j) v(j), \quad i \in \mathscr{X}. \tag{26}$$

By weak duality, $\phi_\infty(\Pi_s, \mathscr{T}_s) \geqslant \psi_\infty(\Pi_s, \mathscr{T}_s)$, where the latter is defined in the theorem. We now prove that equality holds. The lower bound $\psi_\infty(\Pi_s, \mathscr{T}_s)$ can be expressed as the solution to the nonlinear problem (in variables v, τ) obtained by letting the P^as become variables in (25):

$$\psi_\infty(\Pi_s, \mathscr{T}_s) = \max_{\tau \in \mathscr{T}_s,\, v} q^T v:\ v(i) \leqslant c(i, a) + \nu \sum_j P^a(i, j) v(j), \quad a \in \mathscr{A},\ i \in \mathscr{X}. \tag{27}$$

Likewise, if we denote by $\phi_\infty(\pi, \mathscr{T}_s) := \max_{\tau \in \mathscr{T}_s} C_\infty(\pi, \tau)$ the worst-case expected total cost for a given policy π, then this value is obtained by letting the matrices $P^{\mathbf{a}(i)}$ become variables in (14), which results in

$$\phi_\infty(\pi, \mathscr{T}_s) := \max_{\tau \in \mathscr{T}_s, v} q^T v:\ v(i) \leqslant c(i, \mathbf{a}(i)) + \nu \sum_j P^{\mathbf{a}(i)}(i, j) v(j), \quad i \in \mathscr{X}. \tag{28}$$

Due to the rectangular uncertainty property $\mathscr{P}^a = \mathscr{P}_1^a \times \cdots \times \mathscr{P}_n^a$, the problem of computing $\psi_\infty(\Pi_s, \mathscr{T}_s)$, and that of computing $\phi_\infty(\pi, \mathscr{T}_s)$ for a given policy π, can both be represented as problem (24) of Lemma 2, where we define the function g by their components, as follows for problem (27):

$$(g(v))_i := \min_{a \in \mathscr{A}} \big(c(i, a) + \nu \sigma_{\mathscr{P}_i^a}(v)\big), \quad i \in \mathscr{X}, \tag{29}$$

and as follows for problem (28):

$$(g(v))_i := c(i, \mathbf{a}(i)) + \nu \sigma_{\mathscr{P}_i^{\mathbf{a}(i)}}(v^\pi), \quad i \in \mathscr{X}. \tag{30}$$

Because the sets $\mathscr{P}_i^a$ are all included in Δ^n, the above functions are componentwise nondecreasing; furthermore, these functions are ν-contractive, and Lemma 1 applies. This shows that the optimal value of problems (15) and (16) are characterized by the equations given in Theorem 3. The contractive property of g defined by (29) can be established by observing that for any pair $u, v \in \mathbf{R}^n$, and for every $i \in \mathscr{X}$, we have

$$\begin{aligned} g_i(u) &= \min_{a \in \mathscr{A}} \max_{p \in \mathscr{P}_i^a} \big(c(i, a) + \nu p^T v + \nu p^T (u - v)\big) \\ &\leqslant \min_{a \in \mathscr{A}} \max_{p \in \mathscr{P}_i^a} \big(c(i, a) + \nu p^T v\big) + \nu \max_{p \in \mathscr{P}_i^a} p^T (u - v) \\ &\leqslant g_i(v) + \nu \max_{p^T \mathbf{1} = 1,\, p \geqslant 0} p^T (u - v) \\ &\leqslant g_i(v) + \nu \|u - v\|_\infty. \end{aligned}$$

The proof of the contractive property for g defined by (30) is similar. The rest of the proof is similar to that of Theorem 1. This ends our proof. □

Theorem (3) leads to the following theorem.

THEOREM 4. *In the infinite-horizon problem, we can without loss of generality assume that the control and nature policies are stationary, that is,*

$$\phi_\infty(\Pi, \mathscr{T}) = \phi_\infty(\Pi_s, \mathscr{T}_s) = \phi_\infty(\Pi_s, \mathscr{T}) = \phi_\infty(\Pi, \mathscr{T}_s). \tag{31}$$

Furthermore, in the finite-horizon case, with a discounted cost function, the gap between the optimal values of the robust control problems under stationary and time-varying uncertainty models, $\phi_N(\Pi, \mathscr{T}) - \phi_N(\Pi, \mathscr{T}_s)$, goes to zero as the horizon length N goes to infinity, at a geometric rate ν.

PROOF. The proof is in five steps. In Step (a), we prove that $\phi_N(\Pi, \mathscr{T})$ converges to $\phi_\infty(\Pi_s, \mathscr{T}_s)$. In Step (b), we prove that $\Phi_N(\Pi, \mathscr{T})$ converges geometrically at rate ν, to $\phi_\infty(\Pi, \mathscr{T})$, which also proves the first equality in (31). Step (c) proves the second inequality, and Step (d) the last. In Step (d), we prove $\phi_\infty(\Pi, \mathscr{T}_s) = \phi_\infty(\Pi_s, \mathscr{T}_s)$. In Step (e), we prove that $\phi_N(\Pi, \mathscr{T}) - \phi_N(\Pi, \mathscr{T}_s)$ goes to zero as $N \to \infty$, at a geometric rate ν.

Step (a). First, we prove that $\phi_N(\Pi, \mathscr{T})$ converges to $\phi_\infty(\Pi_s, \mathscr{T}_s)$. Denote by (v_k) the iterates of the value function delivered by the infinite-horizon recursion (20), and by v_∞ its limit. We have $\phi_\infty(\Pi_s, \mathscr{T}_s) = v_\infty(i_0)$, where i_0 is the initial state. Fix $\epsilon > 0$; by convergence of recursion (20), there exists a positive integer N_ϵ such that for every $N > N_\epsilon$,

$$\forall\, i \in \mathscr{X}, \quad v_\infty(i) - \epsilon \leqslant v_N(i) \leqslant v_\infty(i). \tag{32}$$

Now fix $N > N_\epsilon$, and define the sequence $\tilde{v}_t = \nu^t v_{N-t}$ for $t = 0, \ldots, N-1$; it satisfies the finite-horizon recursion (7) with the cost function $c_t(i, a) = \nu^t c(i, a)$. Thus, $(\tilde{v}_t)_{t \in T}$ is the optimal value function for the problem of computing $\phi_N(\Pi, \mathscr{T})$, and in particular, $\tilde{v}_0(i_0) = v_N(i_0) = \phi_N(\Pi, \mathscr{T})$. Specializing (32) to $i = i_0$, we obtain

$$\phi_\infty(\Pi_s, \mathscr{T}_s) - \epsilon \leqslant \phi_N(\Pi, \mathscr{T}) \leqslant \phi_\infty(\Pi_s, \mathscr{T}_s), \tag{33}$$

which proves the convergence result.

Step (b). Next, we prove that $\Phi_N(\Pi, \mathscr{T})$ converges geometrically at rate ν, to $\phi_\infty(\Pi, \mathscr{T})$; combining this with Step (a) will then establish the first equality in (31). For every N, the ν-discounted cost function satisfies

$$C_N(\pi, \tau) \leqslant C_\infty(\pi, \tau) \leqslant C_N(\pi, \tau) + \epsilon_N, \tag{34}$$

where $c_{\max} := \max_{i \in \mathscr{X},\, a \in \mathscr{A}} c(i, a) < \infty$ and where $\epsilon_N := \nu^N c_{\max}/(1 - \nu)$ converges geometrically to zero at rate ν. The above implies

$$\phi_N(\Pi, \mathscr{T}) \leqslant \phi_\infty(\Pi, \mathscr{T}) \leqslant \phi_N(\Pi, \mathscr{T}) + \epsilon_N, \tag{35}$$

which in turn proves that $\phi_N(\Pi, \mathscr{T})$ converges geometrically at rate ν, to $\phi_\infty(\Pi, \mathscr{T}) = \phi_\infty(\Pi_s, \mathscr{T}_s)$.

Step (c). To prove $\phi_\infty(\Pi_s, \mathcal{T}_s) = \phi_\infty(\Pi_s, \mathcal{T})$ (the second equality in (31)), we observe that the bounds (34) imply that for every stationary policy π and every N,

$$\phi_N(\pi, \mathcal{T}) \leqslant \phi_\infty(\pi, \mathcal{T}) \leqslant \phi_N(\pi, \mathcal{T}) + \epsilon_N,$$

where $\phi_N(\pi, \mathcal{T}) := \max_{\tau \in \mathcal{T}} C_N(\pi, \tau)$ and $\phi_\infty(\pi, \mathcal{T})$ is defined likewise. This shows that $\lim_{N\to\infty} \phi_N(\pi, \mathcal{T}) = \phi_\infty(\pi, \mathcal{T})$ for every $\pi \in \Pi_s$. Following similar steps as in Step (a), one can prove that for every stationary policy $\pi \in \Pi_s$, $\phi_N(\pi, \mathcal{T})$ converges to $\phi_\infty(\pi, \mathcal{T}_s)$. This ensures that $\phi_\infty(\pi, \mathcal{T}) = \phi_\infty(\pi, \mathcal{T}_s)$ for every $\pi \in \Pi_s$, and hence, $\phi_\infty(\Pi_s, \mathcal{T}) = \phi_\infty(\Pi_s, \mathcal{T}_s)$.

Step (d). To prove the equality

$$\phi_\infty(\Pi, \mathcal{T}_s) = \phi_\infty(\Pi_s, \mathcal{T}_s),$$

we note that standard results on the stationarity of optimal policies for nominal problems (Putterman 1994) imply that for every stationary nature policy $\tau \in \mathcal{T}_s$, $\phi_N(\Pi, \tau)$ converges to $\phi_\infty(\Pi_s, \tau)$ as $N \to \infty$. The equality then follows from the following bound, derived from (34): for every $\tau \in \mathcal{T}_s$, $\phi_N(\Pi, \tau) \leqslant \phi_\infty(\Pi, \tau) \leqslant \phi_N(\Pi, \tau) + \epsilon_N$.

Step (e). Finally, from (34), we obtain

$$\phi_N(\Pi, \mathcal{T}_s) \leqslant \phi_\infty(\Pi, \mathcal{T}_s) \leqslant \phi_N(\Pi, \mathcal{T}_s) + \epsilon_N,$$

which shows that $\phi_N(\Pi, \mathcal{T}_s)$ converges geometrically at a rate ν to $\phi_\infty(\Pi, \mathcal{T}_s) = \phi_\infty(\Pi_s, \mathcal{T}_s) = \phi_\infty(\Pi, \mathcal{T})$. We know from Step (b) that the same holds for $\phi_N(\Pi, \mathcal{T})$, thus the gap $\phi_N(\Pi, \mathcal{T}) - \phi_N(\Pi, \mathcal{T}_s)$ goes to zero as the horizon length N goes to infinity, at a geometric rate ν. □

4.2. Complexity Analysis

We now turn to the complexity analysis of the infinite-horizon problem, assuming again that we use the specific uncertainty models described in §§5 and 6. The robust Bellman recursion (20) provides a sequence (v_k) which converges geometrically at rate ν to the optimal value function v_∞ of the problem. This means that to achieve a given accuracy (say $\epsilon/2$) on that value, we need $O(\log(1/\epsilon))$ iterations, with *exact* computation of the inner problem at each step. Let us examine the complexity when inexact values are used.

THEOREM 5. *For the infinite-horizon problem, if we solve the inner problem with the bisection algorithm accuracy parameter $\delta = (1-\nu)\epsilon/2\nu$, our algorithm will guarantee an ϵ-suboptimal policy, with an additional computational cost of* $\log(1/\epsilon)$ *with respect to the classical dynamic programming algorithm.*

PROOF. We consider iterates $(\hat{v}_k)$ of recursion (20), with the same initial condition $\hat{v}_0 = v_0$, but where we use the bisection algorithm with accuracy $\delta = (1-\nu)\epsilon/2\nu$, in effect replacing the map $\sigma_{\mathcal{P}_i^a}$ by its approximate counterpart $\widehat{\sigma}_{\mathcal{P}_i^a}$, as defined by (18). Let us prove that these approximate values also converge in $O(\log(1/\epsilon))$ time.

We now prove by induction that $v_k \leqslant \hat{v}_k \leqslant v_k + \theta\mathbf{1}$, where $\theta = \nu\delta/(1-\nu) = \epsilon/2$. The initial condition is obtained trivially, as $v_1 = \hat{v}_1$ satisfies $v_1 \leqslant \hat{v}_1 \leqslant v_1 + \theta\mathbf{1}$. Assume that the bounds are true for a given $k \geqslant 1$. Then, for every i, a, we have

$$\begin{aligned}\sigma_{\mathcal{P}_i^a}(v_k) \leqslant \sigma_{\mathcal{P}_i^a}(\hat{v}_k) &\leqslant \sigma_{\mathcal{P}_i^a}(v_k + \theta\mathbf{1}) \leqslant \sigma_{\mathcal{P}_i^a}(v_k) + \sigma_{\mathcal{P}_i^a}(\theta\mathbf{1}) \\ &= \sigma_{\mathcal{P}_i^a}(v_k) + \theta,\end{aligned}$$

where we successively used the convexity, monotonicity, and homogeneity of degree one of the function $\sigma_{\mathcal{P}_i^a}$. We then obtain

$$v_{k+1}(i) \leqslant \hat{v}_{k+1}(i) \leqslant v_{k+1}(i) + \nu(\delta + \theta) = v_{k+1}(i) + \theta \quad \forall i \in \mathcal{X},$$

which proves our result. The above implies that

$$\|\hat{v}_k - v_\infty\|_\infty \leqslant \|\hat{v}_k - v_k\|_\infty + \|v_k - v_\infty\|_\infty \leqslant \theta + \epsilon/2 = \epsilon,$$

provided $k = O(\log(1/\epsilon))$ is large enough. This proves that we can achieve ϵ-convergence of $\hat{v}_k$ in $k = O(\log(1/\epsilon))$.

We finish by examining the cost of computing an ϵ-suboptimal policy. The iterates $\hat{v}_k$ obey to

$$\hat{v}_{k+1}(i) = \min_{a \in \mathcal{A}} \big(c(i,a) + \nu\delta_{\mathcal{P}_i^a}(\hat{v}_k) + \nu\sigma_{\mathcal{P}_i^a}(\hat{v}_k) \big),$$

where $0 \leqslant \delta_{\mathcal{P}_i^a}(\hat{v}_k) \leqslant \delta$. We can express the above as

$$\hat{v}_{k+1}(i) = \min_{a \in \mathcal{A}} \big(c(i,a) + \nu\big(\delta_{\mathcal{P}_i^a}(\hat{v}_k) + \Delta_i^a(k)\big) + \nu\sigma_{\mathcal{P}_i^a}(\hat{v}_{k+1}) \big), \tag{36}$$

where $\Delta_i^a(k) := \sigma_{\mathcal{P}_i^a}(\hat{v}_k) - \sigma_{\mathcal{P}_i^a}(\hat{v}_{k+1})$. $|\Delta_i^a(k)| \leqslant \|\hat{v}_{k+1} - \hat{v}_k\|_\infty$ can be obtained by using the fact that, for any pair of n-vectors (u, v), and subset $\mathcal{P}$ of the probability simplex Δ_n, we have

$$\begin{aligned}\sigma_{\mathcal{P}}(u) - \sigma_{\mathcal{P}}(v) &= \max_{p \in \mathcal{P}} \min_{q \in \mathcal{P}} (p^T u - q^T v) \leqslant \max_{p \in \mathcal{P}} p^T (u - v) \\ &\leqslant \max_{p \in \Delta_n} p^T (u - v) \leqslant \|u - v\|_\infty.\end{aligned}$$

Let $\delta_i^a(k) := \delta_{\mathcal{P}_i^a}(\hat{v}_k) + \Delta_i^a(k)$. Choose $k = N$ such that $\|\hat{v}_{N+1} - \hat{v}_N\|_\infty \leqslant (1-\nu)\epsilon/2\nu$, so that $|\delta_i^a(N)| \leqslant \delta + (1-\nu)\epsilon/2\nu = (1-\nu)\epsilon/\nu$. (By the convergence properties proved above, we have $N = O(\log(1/\epsilon))$.)

Relation (36) implies that $\hat{v}_{k+1}$ and the corresponding policy $\hat{\pi}_k$ is optimal for the infinite-horizon problem, but with a different cost function $\hat{c}$, defined by $\hat{c}(i,a) = c(i,a) + \nu\delta_i^a(N)$. (Note that N is a constant here, so we are really defining a time-invariant cost.) The bound on $\delta_i^a(N)$ then implies that the corresponding expected total discounted cost function satisfies $|\widehat{C}_\infty(\pi, \tau) - C_\infty(\pi, \tau)| \leqslant \epsilon$. The rest of the proof follows that of the finite-horizon case, with the only difference being that now we only have the

two-sided inequality $|C_\infty(\pi,\tau)-\widehat{C}_\infty(\pi,\tau)|\leqslant\epsilon$ as opposed to a one-sided inequality. But the result remains the same.

We established that to compute an ϵ-suboptimal policy, we need to run $O(\log(1/\epsilon))$ steps of the robust Bellman recursion, using a bisection algorithm with accuracy $\delta=O(\epsilon)$. Each step of the Bellman recursion requires $O(mn\log(v_{\max}/\delta))$ flops, which needs to be computed for all the states at each iteration. Hence, the total complexity is $O(mn^2\log(v_{\max}/\delta))$. The bound $v_{\max}\leqslant c_{\max}/(1-\nu)$, where $c_{\max}=\max_{i\in\mathscr{X},\,a\in\mathscr{A}}c(i,a)$, brings the total complexity to $O(mn^2(\log(1/\epsilon))^2)$. Thus, the extra computational cost incurred by robustness in the infinite-horizon case is $O(\log(1/\epsilon))$. □

4.3. Algorithm

Our analysis yields the following algorithm to compute an ϵ-suboptimal policy π^ϵ for problem (6) in $O(mn^2(\log(1/\epsilon))^2)$ flops, using the uncertainty models described in §§5 and 6.

Robust Infinite-Horizon Dynamic Programming Algorithm

Step 1. Set $\epsilon>0$, initialize the value function $\hat{v}_1>0$, and set $k=1$.

Step 2. (a) For all states i and controls a, compute, using the bisection algorithm described in §§5 or 6, a value $\widehat{\sigma}_i^a$ such that

$$\widehat{\sigma}_i^a-\delta\leqslant\sigma_{\mathscr{P}_i^a}(\hat{v}_k)\leqslant\widehat{\sigma}_i^a,$$

where $\delta=(1-\nu)\epsilon/2\nu$.

(b) For all states i and controls a, compute $\hat{v}_{k+1}(i)$ by

$$\hat{v}_{k+1}(i)=\min_{a\in\mathscr{A}}(c(i,a)+\nu\widehat{\sigma}_i^a).$$

Step 3. If

$$\|\hat{v}_{k+1}-\hat{v}_k\|<\frac{(1-\nu)\epsilon}{2\nu},$$

go to Step 4. Otherwise, replace k by $k+1$ and go to Step 2.

Step 4. For each $i\in\mathscr{X}$, set an $\pi^\epsilon=(\mathbf{a}^\epsilon,\mathbf{a}^\epsilon,\ldots)$, where

$$\mathbf{a}^\epsilon(i)=\arg\max_{a\in\mathscr{A}}\{c(i,a)+\nu\widehat{\sigma}_i^a\},\quad i\in\mathscr{X}.$$

5. Likelihood Models

Our first model is based on a likelihood constraint to describe uncertainty on each transition matrix. Our uncertainty model is derived from a controlled experiment starting from state $i=1,2,\ldots,n$ and the count of the number of transitions to different states. We denote by F^a the matrix of empirical frequencies of transition with control a in the experiment; denote by f_i^a its ith row. We have $F^a\geqslant 0$ and $F^a\mathbf{1}=\mathbf{1}$, where $\mathbf{1}$ denotes the vector of ones. For simplicity, we assume that $F^a>0$ for every a.

To simplify the notation, we will drop the superscript a in this section, and refer to a generic transition matrix as P and to its ith row as p_i. The same convention applies to the empirical frequency matrix F^a and its rows f_i^a, as well as to sets $\mathscr{P}^a$ and $\mathscr{P}_i^a$. When the meaning is clear from context, we will further drop the subscript i.

5.1. Model Description

The "plug-in" estimate $\hat{P}=F$ is the solution to the maximum-likelihood problem

$$\max_P L(P):=\sum_{i,j}F(i,j)\log P(i,j):\ P\geqslant 0,\ P\mathbf{1}=\mathbf{1}. \tag{37}$$

The optimal log-likelihood is

$$\beta_{\max}=\sum_{i,j}F(i,j)\cdot\log F(i,j).$$

A classical description of uncertainty in a maximum-likelihood setting is via the likelihood region (Lehmann and Casella 1998, Poor 1988)

$$\left\{P\in\mathbf{R}^{n\times n}:\ P\geqslant 0,\ P\mathbf{1}=\mathbf{1},\ \sum_{i,j}F(i,j)\log P(i,j)\geqslant\beta\right\}, \tag{38}$$

where $\beta<\beta_{\max}$ is a given number, which represents the uncertainty level. In practice, the designer chose an uncertainty level and β can be estimated using resampling methods, or a large sample Gaussian approximation, so as to ensure that the set above achieves the desired level of confidence (see Appendix D).

The above description is classical in the sense that log-likelihood regions are the starting point for developing ellipsoidal or interval models of confidence, hence, are more statistically accurate (Lehmann and Casella 1998); see §7.3 for further details. The above set is statistically meaningful as it describes how informative the data is. If this set is elongated along a direction, then the likelihood function does not vary much in that direction, and the data is not very informative in that direction. This set has some interesting features. First, it does not result from a (quadratic) approximation; it is a valid description of uncertainty, even for β values that are far below $\beta_{\max}$. Second, this set might not be symmetric around the maximum-likelihood point, reflecting the fact the statistical uncertainty depends on the direction. Finally, by construction, it excludes matrices that are not transition matrices; the same cannot be said of the more classical ellipsoidal approximations.

To apply the robust recursion, we need to assume that the uncertainty set $\mathscr{P}$ possesses the rectangular uncertainty property. The likelihood region defined in (38) does not have this property, but we can overapproximate this region by a set that does, by projecting the likelihood regions onto

n-dimensional subspaces, corresponding to the rows of the transition matrix. This overapproximation will result in an upper bound on our optimal control problem, as we are giving more power to nature. Note that this method yields a tighter approximation than that obtained via an interval matrix model, which would require a further overapproximation of the projected sets $\mathcal{P}_i$, by n-dimensional boxes.

Due to the separable nature of the log-likelihood function, the projection of the above set onto the p_i (i.e., row) variables of matrix P can be given explicitly, as

$$\mathcal{P}_i(\beta_i) := \left\{ p \in \Delta^n \colon \sum_j f_i(j) \log p_i(j) \geqslant \beta_i \right\},$$

where

$$\beta_i := \beta - \sum_{k \neq i} \sum_j F(k, j) \log F(k, j).$$

We are now ready to attack problem (17) under the premise that the transition matrix is only known to lie in the rectangular set $\bigotimes_{i=1}^n \mathcal{P}_i(\beta_i)$. The inner problem is to solve an optimization problem of the form

$$\sigma^* := \max_p p^T v \colon\ p \in \Delta^n,\ \ \sum_j f(j) \log p(j) \geqslant \beta, \tag{39}$$

where we have dropped the subscript i in the empirical frequencies vector f_i and in the lower bound β_i. In this section, $\beta_{\max}$ denotes the maximal value of the likelihood function appearing in the above set, which is $\beta_{\max} = \sum_j f(j) \log f(j)$. We assume that $\beta < \beta_{\max}$, which, together with $f > 0$, ensures that the set above has nonempty interior. Without loss of generality, we can assume that $v \in \mathbf{R}^n_+$.

5.2. The Dual Problem

The Lagrangian $\mathcal{L}\colon \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ associated with the inner problem can be written as

$$\mathcal{L}(v, \zeta, \mu, \lambda) = p^T v + \zeta^T p + \mu(1 - p^T \mathbf{1}) + \lambda(f^T \log p - \beta),$$

where ζ, μ, and λ are the Lagrange multipliers. The Lagrange dual function $d\colon \mathbf{R}^n \times \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ is the maximum value of the Lagrangian over p, i.e., for $\zeta \in \mathbf{R}^n$, $\mu \in \mathbf{R}$, and $\lambda \in \mathbf{R}$,

$$\begin{aligned} d(\zeta, \mu, \lambda) &= \sup_p \mathcal{L}(v, \zeta, \mu, \lambda) \\ &= \sup_p \big(p^T v + \zeta^T p + \mu(1 - p^T \mathbf{1}) + \lambda(f^T \log p - \beta) \big). \end{aligned} \tag{40}$$

The optimal $p^* = \arg\sup_p \mathcal{L}(v, \zeta, \mu, \lambda)$ is readily be obtained by solving $\partial \mathcal{L} / \partial p = 0$, which results in

$$p^*(i) = \frac{\lambda f(i)}{\mu - v(i) - \zeta(i)}.$$

Plugging the value of p^* in the equation for $d(\nu, \mu, \lambda)$ yields, with some simplification, the following dual problem:

$$\overline{\sigma} := \min_{\lambda, \mu, \zeta} \mu - (1 + \beta)\lambda + \lambda \sum_j f(j) \log \frac{\lambda f(j)}{\mu - v(j) - \nu(j)} \colon \quad \lambda \geqslant 0, \quad \zeta \geqslant 0, \quad \zeta + v \leqslant \mu \mathbf{1}.$$

Because the above problem is convex, and has a feasible set with nonempty interior, there is no duality gap, that is, $\sigma^* = \overline{\sigma}$. Moreover, by a monotonicity argument, we obtain that the optimal dual variable ζ is zero, which reduces the number of variables to two:

$$\sigma^* = \min_{\lambda, \mu} h(\lambda, \mu),$$

where

$$h(\lambda, \mu) := \begin{cases} \mu - (1 + \beta)\lambda + \lambda \sum_j f(j) \log \dfrac{\lambda f(j)}{\mu - v(j)} \\ \qquad \text{if } \lambda > 0,\ \mu > v_{\max} := \max_j v(j), \\ +\infty \quad \text{otherwise.} \end{cases} \tag{41}$$

For further reference, we note that h is twice differentiable on its domain, and that its gradient is given by

$$\nabla h(\lambda, \mu) = \begin{bmatrix} \sum_j f(j) \log \dfrac{\lambda f(j)}{\mu - v(j)} - \beta \\ 1 - \lambda \sum_j \dfrac{f(j)}{\mu - v(j)} \end{bmatrix}. \tag{42}$$

5.3. A Bisection Algorithm

From the expression of the gradient obtained above, we obtain that the optimal value of λ for a fixed μ, $\lambda(\mu)$, is given analytically by

$$\lambda(\mu) = \left(\sum_j \frac{f(j)}{\mu - v(j)} \right)^{-1}, \tag{43}$$

which further reduces the problem to a one-dimensional problem:

$$\sigma^* = \min_{\mu \geqslant v_{\max}} \sigma(\mu),$$

where $v_{\max} = \max_j v(j)$ and $\sigma(\mu) = h(\lambda(\mu), \mu)$. By construction, the function $\sigma(\mu)$ is convex in its (scalar) argument, because the function h defined in (41) is jointly convex in both its arguments (see Boyd and Vandenberghe 2004, p. 74). Hence, we may use bisection to minimize σ.

To initialize the bisection algorithm, we need upper and lower bounds μ_- and μ_+ on a minimizer of σ. When $\mu \to v_{\max}$, $\sigma(\mu) \to v_{\max}$ and $\sigma'(\mu) \to -\infty$ (see Appendix B). Thus, we may set the lower bound to $\mu_- = v_{\max}$.

The upper bound μ_+ must be chosen such that $\sigma'(\mu_+) > 0$. We have

$$\sigma'(\mu) = \frac{\partial h}{\partial \mu}(\lambda(\mu), \mu) + \frac{\partial h}{\partial \lambda}(\lambda(\mu), \mu)\frac{d\lambda(\mu)}{d\mu}. \tag{44}$$

The first term is zero by construction, and $d\lambda(\mu)/d\mu > 0$ for $\mu > v_{\max}$. Hence, we only need a value of μ for which

$$\frac{\partial h}{\partial \lambda}(\lambda(\mu), \mu) = \sum_j f(j) \log \frac{\lambda(\mu) f(j)}{\mu - v(j)} - \beta > 0. \tag{45}$$

By convexity of the negative log function, and using the fact that $f^T \mathbf{1} = 1$, $f \geqslant 0$, we obtain that

$$\begin{aligned}\frac{\partial h}{\partial \lambda}(\lambda(\mu), \mu) &= \beta_{\max} - \beta + \sum_j f(j) \log \frac{\lambda(\mu)}{\mu - v(j)} \\ &\geqslant \beta_{\max} - \beta - \log\left(\sum_j f(j) \frac{\mu - v(j)}{\lambda(\mu)}\right) \\ &\geqslant \beta_{\max} - \beta + \log \frac{\lambda(\mu)}{\mu - \bar{v}},\end{aligned}$$

where $\bar{v} = f^T v$ denotes the average of v under f.

The above, combined with the bound on $\lambda(\mu)$: $\lambda(\mu) \geqslant \mu - v_{\max}$, yields a sufficient condition for (45) to hold:

$$\mu > \mu_+^0 := \frac{v_{\max} - e^{\beta - \beta_{\max}} \bar{v}}{1 - e^{\beta - \beta_{\max}}}. \tag{46}$$

By construction, the interval $[v_{\max}, \mu_+^0]$ is guaranteed to contain a global minimizer of σ over $(v_{\max}, +\infty)$.

The bisection algorithm is as follows:

Step 1. Set $\mu_- = v_{\max}$ and $\mu_+ = \mu_+^0$ as in (46). Let $\delta > 0$ be a small convergence parameter.

Step 2. While $\mu_+ - \mu_- > \delta(1 + \mu_- + \mu_-)$, repeat

(a) Set $\mu = (\mu_+ + \mu_-)/2$.

(b) Compute the gradient of σ at μ.

(c) If $\sigma'(\mu) > 0$, set $\mu_+ = \mu$; otherwise, set $\mu_- = \mu$.

(d) go to 2a.

In practice, the function to minimize may be very "flat" near the minimum. This means that the above bisection algorithm may take a long time to converge to the global minimizer. Because we are only interested in the value of the minimum (and not of the minimizer), we may modify the stopping criterion to

$$\mu_+ - \mu_- \leqslant \delta(1 + \mu_- + \mu_-) \quad \text{or} \quad \sigma'(\mu_+) - \sigma'(\mu_-) \leqslant \delta.$$

The second condition in the criterion implies that

$$|\sigma'((\mu_+ + \mu_-)/2)| \leqslant \delta,$$

which is an approximate condition for global optimality.

5.4. Complexity

Let us analyze the number of iterations needed to achieve a given accuracy on the optimal value σ^*. We denote by μ^* a minimizer of the function and by μ_+, μ_- the final iterates of the bisection algorithm, run with convergence parameter δ. We then have $\mu_+ - \mu_- \leqslant \delta(1 + 2\mu_+^0)$, which implies

$$0 \leqslant \mu_+ - \mu^* \leqslant \mu_+ - \mu_- \leqslant \delta\left(1 + \frac{2v_{\max}}{1 - e^{\beta - \beta_{\max}}}\right) = O(v_{\max}\delta).$$

The number of iterations needed to achieve the above bound on the minimizer μ^* grows as $\log((\mu_+^0 - v_{\max})/\delta) = O(\log(v_{\max}/\delta))$. Thus, to achieve an accuracy δ in the minimizer, we need $O(\log(v_{\max}/\delta))$ iterations.

Here, we are not interested in the value of a minimizer μ^*, but on the minimum value, σ^*. By construction, $\mu_+ \geqslant \mu^*$, and we have $0 \leqslant \sigma'(\mu_+) \leqslant \lim_{\mu \to +\infty} \sigma'(\mu) = \beta_{\max} - \beta$. Furthermore, we have $0 \leqslant \mu_+ - \mu^* \leqslant \mu_+ - \mu_- \leqslant O(v_{\max}\delta)$. By convexity,

$$\begin{aligned}\sigma(\mu_+) &\geqslant \sigma^* \geqslant \sigma(\mu_+) - (\mu_+ - \mu^*)\sigma'(\mu_+) \\ &= \sigma(\mu_+) - O(v_{\max}\delta).\end{aligned}$$

We obtain that, to achieve a given accuracy δ on σ^*, we need $O(\log(v_{\max}/\delta))$ iterations of the bisection algorithm. Because each iteration requires n flops, the total complexity of the inner problem is $O(n \log(v_{\max}/\delta))$.

5.5. Maximum A Posteriori Models

We now consider a variation on the likelihood model, the maximum a posteriori (MAP) model. The MAP estimation framework provides a way of incorporating prior information in the estimation process. This is particularly useful for dealing with sparse training data, for which the maximum-likelihood approach may provide inaccurate estimates. The MAP estimator, denoted by p^{MAP}, maximizes the "MAP function" (Siouris 1995)

$$L_{\text{MAP}}(p) = L(p) + \log g_{\text{prior}}(p),$$

where $L(p)$ is the log-likelihood function, and g_{prior} refers to the a priori density function of the parameter vector p.

In our case, p is a row of the transition matrix, so a prior distribution has support included in the n-dimensional simplex $\{p\colon p \geqslant 0,\ p^T \mathbf{1} = 1\}$. It is customary to choose the prior to be a Dirichlet distribution (Ferguson 1974, Wilks 1962), the density of which is of the form

$$g_{\text{prior}}(p) = K \cdot \prod_i p_i^{\alpha_i - 1},$$

where the vector $\alpha \geqslant \mathbf{1}$ is given and K is a normalizing constant. Choosing $\alpha = \mathbf{1}$, we recover the "noninformative prior," which is the uniform distribution on the n-dimensional simplex. In that case, the MAP estimation converges to the maximum-likelihood estimation. Hence,

the MAP estimation is a more general framework and the maximum-likelihood estimation is a specialization of the MAP when prior information is not available.

The resulting MAP estimation problem takes the form

$$\max_p (f+\alpha-\mathbf{1})^T \log p\colon\ p^T\mathbf{1}=1,\quad p\geqslant 0.$$

To this problem, we can associate a MAP region that describes the uncertainty on the estimate, via a lower bound β on the function $L_{\mathrm{MAP}}(p)$. The inner problem now takes the form

$$\sigma := \max_p p^T v\colon\ p\geqslant 0,\quad p^T\mathbf{1}=1,$$
$$\sum_j (f(j)+\alpha(j)-1)\log p(j)\geqslant \kappa,$$

where κ depends on the normalizing constant K appearing in the prior density function and on the chosen lower bound on the MAP function, β. We observe that this problem has exactly the same form as in the case of the likelihood function, provided we replace f by $f+\alpha-\mathbf{1}$. Therefore, the same results apply to the MAP case.

6. Entropy Models

6.1. Model Description

We consider problem (17), with the uncertainty on the ith row of the transition matrix P^a described by a set of the form $\mathcal{P}=\{p\in\Delta_n\colon\ D(p\|q)\leqslant\beta\}$, where $\beta>0$ is fixed, $q>0$ is a given distribution, and $D(p\|q)$ denotes the Kullback-Leibler divergence from q to p:

$$D(p\|q) := \sum_j p(j)\log\frac{p(j)}{q(j)}.$$

Together with $q>0$, the condition $\beta>0$ ensures that $\mathcal{P}$ has nonempty interior. (As before, we have dropped the control and row indices a and i.)

Note that both the likelihood and entropy models can be interpreted in terms of an upper bound on the Kullback-Leibler divergence between two distributions. In the likelihood setting, we impose an upper bound on the divergence $D(f\|p)$, from the (unknown) distribution p to the observed distribution f; in the entropy case, we use an upper bound on the divergence from the reference distribution q to the unknown distribution p. This parallel suggests a heuristic to choose the uncertainty level β by following the same guidelines used in the likelihood setting, as described in Appendix D.

We now address the inner problem (17), with $\mathcal{P}$ given above. We note that $\mathcal{P}$ actually equals the whole probability simplex if β is too large, specifically if $\beta\geqslant\max_i(-\log q_i)$, because the latter quantity is the maximum of the relative entropy function over the simplex. Thus, if $\beta\geqslant\max_i(-\log q_i)$, the worst-case value of $p^T v$ for $p\in\mathcal{P}$ is equal to $v_{\max}:=\max_j v(j)$.

6.2. Dual Problem

By standard duality arguments (set $\mathcal{P}$ being of nonempty interior), the inner problem is equivalent to its dual:

$$\min_{\lambda>0,\,\mu}\ \mu+\beta\lambda+\lambda\sum_j q(j)\exp\left(\frac{v(j)-\mu}{\lambda}-1\right).$$

Setting the derivative with respect to μ to zero, we obtain the optimality condition

$$\sum_j q(j)\exp\left(\frac{v(j)-\mu}{\lambda}-1\right)=1,$$

from which we derive

$$\mu=\lambda\log\left(\sum_j q(j)\exp\frac{v(j)}{\lambda}\right)-\lambda.$$

The optimal distribution is

$$p^*=\frac{q(j)\exp(v(j)/\lambda)}{\sum_i q(i)\exp(v(i)/\lambda)}.$$

As before, we reduce the problem to a one-dimensional problem

$$\min_{\lambda>0}\sigma(\lambda),$$

where σ is the convex function

$$\sigma(\lambda)=\lambda\log\left(\sum_j q(j)\exp\frac{v(j)}{\lambda}\right)+\beta\lambda. \tag{47}$$

Perhaps not surprisingly, the above function is closely linked to the moment-generating function of a random variable $\mathbf{v}$ having the discrete distribution with mass q_i at v_i.

6.3. Bisection Algorithm

As proved in Appendix C, the convex function σ in (47) has the following properties:

$$\forall\,\lambda\geqslant 0,\quad q^T v+\beta\lambda\leqslant\sigma(\lambda)\leqslant v_{\max}+\beta\lambda \tag{48}$$

and

$$\sigma(\lambda)=v_{\max}+(\beta+\log Q(v))\lambda+o(\lambda), \tag{49}$$

where

$$Q(v) := \sum_{j:v(j)=v_{\max}} q(j)=\mathbf{Prob}\{\mathbf{v}=v_{\max}\}.$$

Hence, $\sigma(0)=v_{\max}$ and $\sigma'(0)=\beta+\log Q(v)$. In addition, at infinity the expansion of σ is

$$\sigma(\lambda)=q^T v+\beta\lambda+o(1). \tag{50}$$

The bisection algorithm can be started with the lower bound $\lambda_-=0$. An upper bound can be computed by finding a solution to the equations $\sigma(0)=q^T v+\beta\lambda$, which yields

the initial upper bound $\lambda_+^0 = (v_{\max} - q^T v)/\beta$. By convexity, a minimizer exists in the interval $[0\ \lambda_+^0]$.

Note that if $\sigma'(0) \geqslant 0$, then $\lambda = 0$ is optimal and the optimal value of σ is $v_{\max}$. This means that if β is too high, that is, if $\beta > -\log Q(v)$, enforcing robustness amounts to disregard any prior information on the probability distribution p. We have observed in §6.1 a similar phenomenon brought about by too large values of β, which resulted in a set $\mathcal{P}$ equal to the probability simplex. Here, the limiting value $-\log Q(v)$ depends not only on q but also on v, because we are dealing with the optimization problem (17) and not only with its feasible set $\mathcal{P}$.

6.4. Complexity

The complexity analysis for the entropy model follows the same lines as that of the likelihood model, so we will be brief here. First, we note that the number of iterations needed to obtain a given accuracy δ on the minimizer is $O(\log(v_{\max}/\delta))$ iterations, because $\lambda_+^0 = O(v_{\max})$. To obtain a given accuracy on the minimum value, the important feature is to ensure that the derivative of the function σ is bounded uniformly and independent of problem size n, at least on one side of the optimum. In the entropy case, we have at each step of the bisection algorithm $0 \leqslant \sigma'(\mu_+) \leqslant \lim_{\mu\to+\infty} \sigma'(\mu) = \beta$. We obtain that, to achieve a given accuracy δ on σ^*, we need $O(\log(v_{\max}/\delta))$ iterations of the bisection algorithm. Because each iteration requires n flops, the total complexity of the inner problem in the entropy case is again $O(n\log(v_{\max}/\delta))$.

7. Other Uncertainty Models

7.1. Finite Scenario Model

Perhaps the simplest uncertainty model involves a finite collection of transition matrices, where for every $a \in \mathcal{A}$, $\mathcal{P}^a = \{P^{a,1}, \ldots, \mathcal{P}^{a,L}\}$, with $P^{a,k} \in \Theta_n$ representing a possible value (scenario) of the transition matrix. As noted earlier, the robust Bellman recursion applies to nonconvex uncertainty sets $\mathcal{P}^a$, as long as they satisfy the rectangular uncertainty property, which is certainly the case here. Note that the scenario model gives rise to the same optimal robust policy as when the finite set $\mathcal{P}^a$ above is replaced by a product of convex hulls: $\bigotimes_{i=1}^n \mathbf{conv}\{p_i^{a,1}, \ldots, p_i^{a,L}\}$, where $p_i^{a,k}$ denotes the ith row of matrix $P^{a,k}$.

Under the scenario (or polytopic) model, the inner problem (17) bears a particularly simple form:

$$\sigma_{\mathcal{P}_i^a}(v) = \max_{p\in\{p_i^{a,1},\ldots,p_i^{a,L}\}} v^T p = \max_{1\leqslant k\leqslant L}, \quad v^T p_i^{a,k}.$$

The worst-case complexity of each step of the robust Bellman recursion is then $O(mnL)$, where L is the number of vertices. For moderately large values of L, the scenario model is attractive, due to its simplicity of implementation.

7.2. Interval Matrix Model

The interval matrix model describes the uncertainty on the rows of the transition matrices in the form

$$\mathcal{P} = \{p:\ \underline{p} \leqslant p \leqslant \bar{p},\ p^T\mathbf{1} = 1\},$$

where $\bar{p}$, $\underline{p}$ are given componentwise nonnegative n-vectors (whose elements do not necessarily sum to one), with $\bar{p} \geqslant \underline{p}$. Note that for Theorem 1 to hold, we must ensure that the set $\mathcal{P}$ is entirely included in the probability simplex Δ_n, which we did by assuming $\underline{p} \geqslant 0$. This model is motivated by statistical estimates of intervals of confidence on the *components* of the transition matrix. Those intervals can be obtained by resampling methods, or by projecting an ellipsoidal uncertainty model on each component axis (see §7.3). Because $\bar{p} \geqslant \underline{p}$, $\mathcal{P}$ is not empty.

Because the inner problem

$$\sigma^* := \max_p v^T p:\ p^T\mathbf{1} = 1, \quad \underline{p} \leqslant p \leqslant \bar{p}$$

is a linear, feasible program, it is equivalent to its dual, which can be reduced to

$$\sigma^* = \min_\mu (\bar{p} - \underline{p})^T(\mu\mathbf{1} - v)^+ + v^T\bar{p} + \mu(1 - \bar{p}^T\mathbf{1}),$$

where z^+ stands for the positive part of vector z. The function to be minimized is a convex piecewise linear function with break points $v(0) := 0$ and $v(1), \ldots, v(n)$. Because the original problem is feasible, we have $\mathbf{1}^T\underline{p} \leqslant 1$, which implies that the function above goes to infinity when $\mu \to \infty$. Thus, the minimum of the function is attained at one of the break points $v(i)$ $(i = 0, \ldots, n)$. The complexity of this enumerative approach is $O(n^2)$, because each evaluation costs $O(n)$. In fact, one does not need to enumerate the function at all values v_i; a bisection scheme over the discrete set $\{v_0, \ldots, v_n\}$ suffices. This scheme will bring the complexity down to $O(n\log n)$.

7.3. Ellipsoidal Models

Ellipsoidal models arise when second-order approximations are made to the log-likelihood function arising in the likelihood model. Specifically, we work with the following set in lieu of (38):

$$\mathcal{P}(\beta) = \left\{P \in \mathbf{R}^{n\times n}:\ P \geqslant 0,\ P\mathbf{1} = \mathbf{1},\ Q(P) \geqslant \beta\right\}, \tag{51}$$

where $Q(P)$ is the second-order approximation to the log-likelihood function L, around the maximum-likelihood estimate F:

$$Q(P) := \beta_{\max} - \frac{1}{2}\sum_{i,j}\frac{(P(i,j) - F(i,j))^2}{F(i,j)}.$$

The above set is an ellipsoid intersected by the polytope of transition matrices. Again, to ensure the rectangular uncertainty property, we first form the projections on the space

of ith row variables. These assume a similar shape, that of an ellipsoid intersected with the probability simplex, specifically,

$$\mathcal{P}_i(\beta) = \left\{ p\colon\ p \geqslant 0,\ p^T\mathbf{1} = 1,\ \sum \frac{(p_i(j) - f_i(j))^2}{f_i(j)} \leqslant \kappa^2 \right\},$$

where $\kappa^2 := 2(\beta_{\max} - \beta)$. We refer to the above model as the *constrained ellipsoidal model*.

In the constrained likelihood case, the inner problem assumes the form

$$\max_p v^T p\colon\ p \geqslant 0,\ \ p^T\mathbf{1} = 1,\ \ \sum \frac{(p(j) - f(j))^2}{f(j)} \leqslant \kappa^2.$$

Using an interior-point method (Boyd and Vandenberghe 2004), the above problem can be solved with absolute accuracy ϵ in worst-case time of $O(n^{1.5}\log(v_{\max}/\epsilon))$, and with a *practical* complexity of $O(n\log(v_{\max}/\epsilon))$.

In statistics, it is a standard practice to further simplify the description above, by relaxing the inequality constraints $P \geqslant 0$ in the definition of $\mathcal{P}(\beta)$. This would bring down the worst-case complexity to $O(n\log(v_{\max}/\epsilon))$. However, if sign constraints are omitted, Theorem 1 does not necessarily hold, and we would only compute an upper bound on the value of the problem.

8. Example: Robust Aircraft Routing

We consider the problem of routing an aircraft whose path is obstructed by stochastic obstacles, representing storms. In practice, the stochastic model must be estimated from past weather data. This makes this particular application a good illustration of our method.

8.1. The Nominal Problem

In Nilim et al. (2001), we introduce an MDP representation of the problem, in which the evolution of the storms is modelled as a *perfectly* known stationary Markov chain. The term nominal here refers to the fact that the transition matrix of the Markov process corresponding to the weather is not subject to uncertainty. The goal is to minimize the expected delay (flight time). The weather process is a fully observable Markov chain: At each decision stage (every 15 minutes in our example), we learn the actual state of the weather.

The air space is represented as a rectangular grid. The state vector comprises the current position of the aircraft on the grid, as well as the current states of each storm. The action in the MDP corresponds to the choice of nodes to fly towards, from any given node. There are k obstacles, represented by a Markov chain with a $2^k \times 2^k$ transition matrix. The transition matrix for the routing problem is thus of order $N2^k$, where N is the number of nodes in the grid.

We solved the MDP via the Bellman recursion (Nilim et al. 2001). Our framework avoids the potential "curse of dimensionality" inherent in generic Bellman recursions, by considerable pruning of the state space and action sets. This makes the method effective for up to a few storms, which corresponds to realistic situations. For more details on the nominal problem and its implementation, we refer the reader to Nilim et al. (2001).

In the example below, the problem is two-dimensional in the sense that the aircraft flies at a fixed altitude. In a coordinate system where each unit is equal to 1 nautical mile, the aircraft is initially positioned at $(0, 0)$ and the destination point is at $(360, 0)$. The velocity of the aircraft is fixed at 480 n.mi/hour. The air space is described by a rectangular grid with $N = 210$ nodes, with edge length of 24 n.mi. There is a possibility that a storm might obstruct the flight path. The storm zone is a rectangular space with the corner points at $(160, 192)$, $(160, -192)$, $(168, 192)$, and $(168, -192)$ (Figure 1).

Because there is only one potential storm in the area, storm dynamics is described by a 2×2 transition matrix P_{weather}. Together with $N = 210$ nodes, this results in a state space of total dimension 420. By limiting the angular changes in the heading of the aircraft, we can prune out the action space and reduce its cardinality at each step to $m = 4$. This implies that the transition matrices are very sparse; in fact, they are sparse, affine functions of the transition matrix P_{weather}. Sparsity implies that the nominal Bellman recursion only involves 8 states at each step.

8.2. The Robust Version

In practice, the transition matrix P_{weather} is estimated from past weather data, and thus it is subject to estimation errors.

We assumed a likelihood model of uncertainty on this transition matrix. This results in a likelihood model of uncertainty on the state transition matrix, which is as sparse as the nominal transition matrix. Thus, the effective state pruning that takes place in the nominal model can also take

Figure 1. Aircraft path planning scenario.

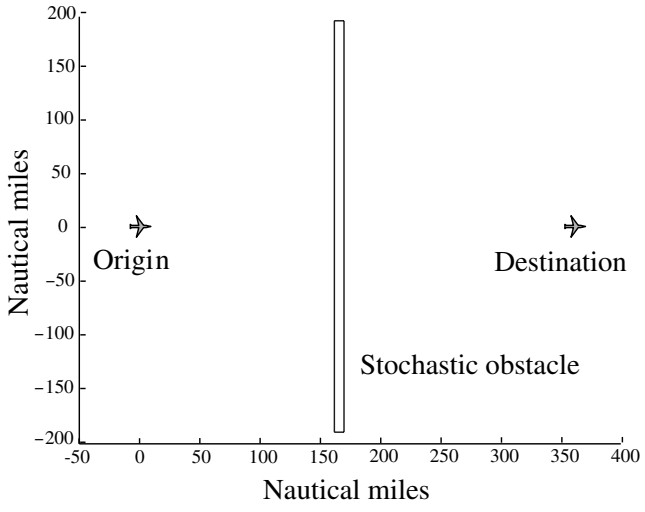
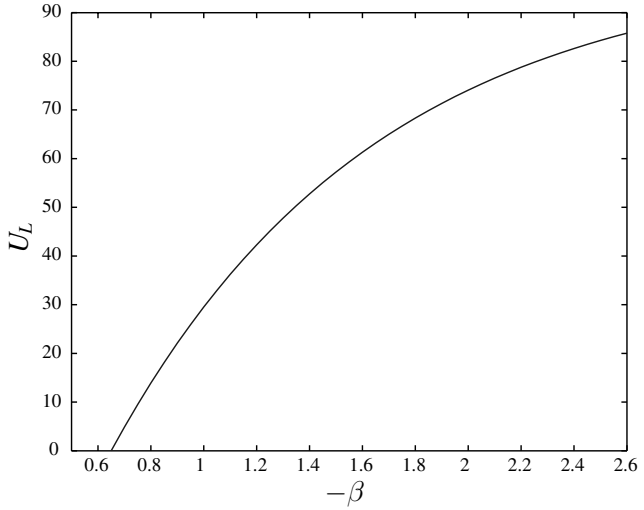


Figure 2. $-\beta$ (negative lower bound on the log-likelihood function) vs. U_L (uncertainty level in % of the transition matrices).



place in the robust counterpart. In our example, we chose the numerical value

$$P_{\text{weather}} = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$

for the maximum-likelihood estimate of P_{weather}.

The likelihood model involves a lower bound β on the likelihood function, which is a measure of the uncertainty level. Its maximum value $\beta_{\max}$ corresponds to the case with no uncertainty, and decreasing values of β correspond to a higher uncertainty level. To β, we may associate a measure of uncertainty that is perhaps more readable: The *uncertainty level*, denoted by U_L, is defined as a percentage and its complement $1 - U_L$ can be interpreted as a probabilistic confidence level in the context of large samples. The one-to-one correspondence of U_L and β is precisely described in Appendix D.

In Figure 2, we plot U_L against decreasing values of the lower bound on the log-likelihood function (β). We see that $U_L = 0$, which refers to a complete certainty of the data, is attained at $\beta = \beta_{\max}$, the maximum value of the likelihood function. The value of U_L decreases with β and reaches the maximum value, which is 100%, at $\beta = -\infty$ (not drawn in this plot). Point to be noted: The rate of increase of U_L is maximum at $\beta = \beta_{\max}$ and increases with β.

8.3. Comparing Robust and Nominal Strategies

In Figure 3, we compare various strategies: We plot the relative delay, which is the relative increase (in percentage) in flight time with respect to the flight time corresponding to the most direct route (straight line), against the negative of the lower bound on the likelihood function β.

We compare three strategies. The *conservative strategy* is to avoid the storm zone altogether. If we take $\beta = \beta_{\max}$, the uncertainty set becomes a singleton ($U_L = 0$) and hence we obtain the solution computed via the classical Bellman recursion; this is referred to as the *nominal strategy*. The *robust strategy* corresponds to solving our robust MDP with the corresponding value of β.

The plot in Figure 3 shows how the various strategies fare, as we decrease the bound on the likelihood function β. For the nominal and the robust strategies, and a given bound β, we can compute the worst-case delay using recursion (10), which provides the worst-case value function.

The conservative strategy incurs a 51.5% delay with respect to the flight time corresponding to the most direct route. This strategy is independent of the transition matrix, so it appears as a straight line in the plot. If we know the value of the transition matrix exactly, then the nominal strategy is extremely efficient and results in a delay of 8.02% only. As β deviates from $\beta_{\max}$, the uncertainty set gets bigger. In the nominal strategy, the optimal value is very sensitive in the range of values of β close to $\beta_{\max}$: the delay jumps from 8% to 25% when β changes by 7.71% with respect to $\beta_{\max}$ (the uncertainty level U_L changes from 0% to 5%). In comparison, the relative delay jumps by only 6% with the robust strategy. In both strategies, the slope of the optimal value with respect to the uncertainty is almost infinite at $\beta = \beta_{\max}$, which shows the high sensitivity of the value function with respect to the uncertainty.

We observe that the robust solution performs better than the nominal solution as the estimation error increases. The plot shows an average of 19% decrease in delay with respect to the nominal strategy when uncertainty is present. Further, as the uncertainty level increases, the nominal strategy very quickly reaches delay values comparable to those obtained with the conservative strategy. In fact, the conservative strategy even outperforms the nominal strategy at $\beta = -1.84$, which corresponds to $U_L = 69.59\%$. In this sense, even for moderate uncertainty levels, the nominal

Figure 3. Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function) for the classical Bellman recursion and its robust counterpart.

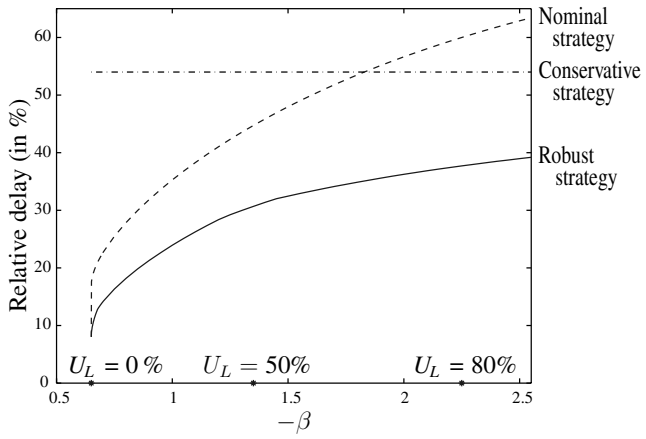
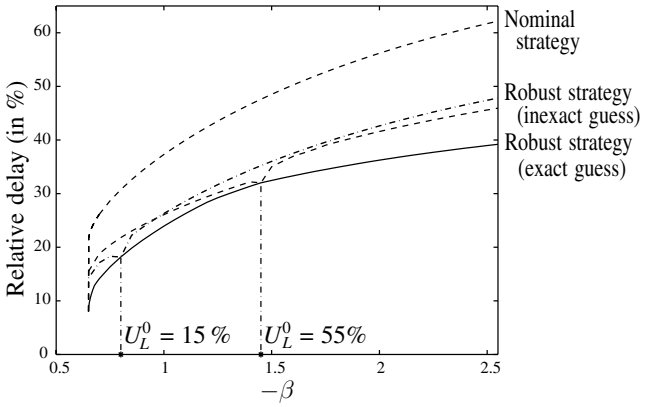


Figure 4. Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function) for the classical Bellman recursion and its robust counterpart (with exact and inexact predictions of the uncertainty level U_L).



strategy defeats its purpose. In contrast, the robust strategy outperforms the conservative strategy by 15% even if the data is very uncertain ($U_L = 85\%$).

In summary, when there is no error in the estimation, both nominal and robust algorithms provide a strategy that produces 43.3% less delay than the conservative strategy. However, with the presence of even a moderate estimation error, the robust strategy performs much better than the conservative strategy, whereas the nominal MDP strategy cannot produce a much better result.

Nominal and robust strategies have similar computational requirements. In our example, with a simple Matlab implementation on a standard PC, the running time for the nominal algorithm was about four seconds, and the robust version took on average four more seconds to solve.

8.4. Inaccuracy of Uncertainty Level

The previous comparison assumes that in the robust case, we are able to estimate exactly the precise value of the uncertainty level U_L (or the bound on the likelihood function β). In practice, this parameter also has to be estimated. Hence the question: How sensitive is the robust approach with respect to inaccuracies in the uncertainty level U_L?

To answer this question in our particular example, we have assumed that a guess U_L^0 on the uncertainty level is available, and examined how the corresponding robust solution would behave if it was subject to uncertainty with level above or below the guess.

In Figure 4, we compare various strategies. In each strategy, we guess a desired level of accuracy (U_L^0) on the data and calculate a corresponding likelihood bound β^0. We choose the optimal action using our robust MDP algorithm applied with this bound. Keeping the resulting policy fixed, we then compute the relative delay with the various values of β. In Figure 4, we plot the relative delays against $-\beta$ for the strategies where the uncertainty levels were guessed as 15% and 55%.

Not surprisingly, the relative delay of a strategy attains its minimum value when β (U_L) is accurately predicted. For values of β above or below its guessed value, the delay increases. We note that it is only for very small uncertainty levels (within 0.995% of $\beta_{\max}$) that the nominal strategy performs better than the robust strategy with imperfect prediction of β (U_L).

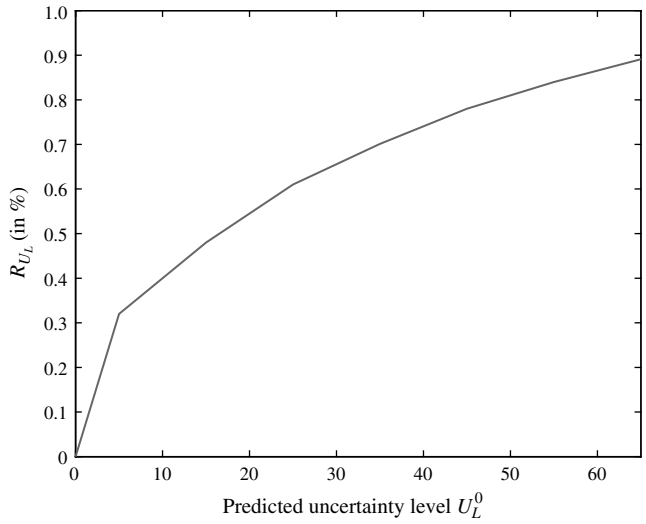
We define R_{U_L} as the range of the actual U_L in percentage terms, where the robust strategy (with imperfect prediction of U_L) performs worse than nominal strategy. In Figure 5, we show R_{U_L} against the guessed value, U_L^0. The plot clearly shows that R_{U_L} remains less than 1% with varying predicted U_L^0.

Our example shows that if we predict the uncertainty level inaccurately to obtain a robust strategy, the nominal strategy will outperform the robust strategy only if the actual uncertainty level U_L is less than 1%. For any higher value of the uncertainty level, the robust strategies outperform the nominal strategy by an average of 13%. Thus, even if the uncertainty level is not accurately predicted, the robust solution outperforms the nominal solution significantly.

9. Concluding Remarks

We have considered a robust Markov decision problem with uncertainty models for the transition matrices that are statistically accurate, yet give rise to very moderate extra computational effort for computing a robust solution, with respect to a nominal solution, where uncertainty is ignored. Specifically, the relative increase in computational cost is of order

Figure 5. Predicted uncertainty level U_L^0 vs. R_{U_L}, which is the range of the actual uncertainty level U_L over which the nominal strategy performs better than a robust strategy computed with the imperfect prediction U_L^0.



$O(\log(N/\epsilon))$ in the finite-horizon case, and $O(\log(1/\epsilon))$ in the infinite-horizon case, where ϵ is the desired accuracy on the optimal expected total cost. As a result, the robust algorithm has practically the same complexity as that of the nominal problem. We have considered both stationary and time-varying assumptions about uncertainty, and showed that as the decision horizon goes to infinity, the gap between these two models vanishes. This justifies our use of bounds based on stationarity assumptions, even if we allow time-varying changes in the transition matrices. The statistical accuracy of our uncertainty models is derived from the fact that they use the Kullback-Leibler divergence, which is a natural way to measure errors in the transition matrices. The other models we have considered, from the polytopic to the interval to the ellipsoidal model, do not enjoy such properties, and moreover, give rise to larger worst-case complexity estimates.

We have shown in a practical path planning example the benefits of using a robust strategy instead of the classical optimal strategy; even if the uncertainty level is only crudely guessed, the robust strategy yields a much better worst-case expected flight delay.

Appendix A. Stochastic Game-Theoretic Proof of the Robust Bellman Recursion

In this section, we prove that the stochastic game with perfect information (4) can be solved using the robust Bellman recursion (7). Our proof is based on transforming the original problem into a term-based zero-sum game, and applying a result by Nowak (Altman et al. 2000, Altman and Hordijk 1994, Nowak 1984) that applies to such games.

We begin by augmenting the state space $\mathscr{X}$ with states of the form (i, a), where $i \in \mathscr{X}$ and $a \in \mathscr{A}$. The augmented state space is thus $\mathscr{X}^{\text{aug}} := \mathscr{X} \cup (\mathscr{X} \times \mathscr{A})$. We now define a new two-player game on this augmented state space, where decisions are taken not only at time t, $t \in T = \{0, 1, \ldots, N\}$, but also at intermediate times $t + 1/2$, $t \in T$.

In the first step, from t to $t + 1/2$, if the system is in a state of the form i, a deterministic a_t results in a transition to the state (i, a_t) with probability one, and the incurred cost is the cost of the original problem, $c_t(i, a_t)$. If the system is in a state of the form (i, a_t), then the controller is not allowed to choose any action and the states stay the same with probability one; the incurred cost in this case is zero. Randomized actions of the controller can be described by a probability measure $\mathbf{q} \in \Delta_m$ (the probability simplex in $\mathbf{R}^m$). In the first step, the opponent is idle.

In the second step, from $t + 1/2$ to $t + 1$, the controller stands idle while the opponent acts as follows. The states of the form (i, a) make a transition to states of the form j with probability $p_i^a(j)$, where p_i^a is freely chosen by the opponent from the set $\mathscr{P}_i^a$. If the system is at any state of the form i at $t + 1/2$, it remains at the same state with probability one. There is no cost incurred at this stage.

Clearly, starting at time t in state i, and with a controller action a, we end up in the state j at time $(t + 1)$ with probability $p_i^a(j)$. Because incurred costs are the same, our new game is equivalent to the original game. In addition, the new game is a term-based zero-sum game, because the controller and the opponent act alternatively, in an independent fashion at each time step. Note that the rectangular uncertainty property is crucial here, as it ensures the fact that the opponent is free to chose p_i^a in the set $\mathscr{P}_i^a$.

Nowak's result provides a Bellman-type recursion to solve the problem of minimizing the worst-case (maximum) expected cost of a term-based zero-sum game, when both players follow randomized policies that are restricted to given state-dependent compact subsets of the probability simplex. In our new game, the opponent's choice of a vector p_i^a within $\mathscr{P}_i^a$ at the second step, can be interpreted as a choice of a randomized policy over the compact, convex, state-dependent set $\mathscr{B}((i, a)) := \mathbf{conv}(\mathscr{P}_i^a)$, the convex hull of the set $\mathscr{P}_i^a$. This ensures that the set of transition measures is convex. (Here, the deterministic actions of the opponent correspond to the vertices of the probability simplex of $\mathbf{R}^n$.) Hence, the results due to Nowak (1984) apply.

In the case when both of the players choose the randomized, state-independent actions, the recursion for the optimal value function v_k in state s can be written for $k = 0, 1/2, 1, \ldots, N - 1/2$, as

$$v_k(s) = \min_{\mathbf{q} \in \Delta_m} \max_{\mathbf{b} \in \mathscr{B}(s)} \mathbf{E}_{\mathbf{qb}}(c_k(s, a, b) + v_{k+1/2}(s')) \quad \forall\, s' \in \mathscr{X}^{\text{aug}}, \tag{52}$$

where the notation c_k is the cost function, $\mathbf{q}$ refers to a particular randomized action of the controller that is freely chosen by the controller from Δ_m, $\mathbf{b}$ refers to a particular randomized action that is freely chosen by the opponent within the state-dependent compact set $\mathscr{B}(s) = \mathbf{conv}(\mathscr{P}_i^a)$, and $\mathbf{E}_{\mathbf{qb}}$ is the corresponding expectation operator with respect to the product measure $\mathbf{q} \otimes \mathbf{b}$. The boundary condition of the game is $v_N(s) = c_N(s)\ \forall\, s \in \mathscr{X} \subset \mathscr{X}^{\text{aug}}$. Due to the sequential nature of the game, (52) can be rewritten as

$$v_k(s) = \min_{\mathbf{q} \in \Delta_m} \mathbf{E}_{\mathbf{q}}\Big(c_k(s, a, b) + \max_{\mathbf{b} \in \mathscr{B}(s)} \mathbf{E}_{\mathbf{b}}(v_{k+1}(s'))\Big). \tag{53}$$

Because, $\mathbf{E}_{\mathbf{b}}(v_{k+1})$ is a linear function of the measure $\mathbf{b}$, it can be easily shown that

$$\max_{\mathbf{b} \in \mathscr{B}(s) = \mathbf{conv}(\mathscr{P}_i^a)} \mathbf{E}_{\mathbf{b}}(v_{k+1}) = \max_{\mathbf{b} \in \mathscr{P}_i^a} \mathbf{E}_{\mathbf{b}}(v_{k+1}). \tag{54}$$

Let us detail how applying the above recursion to our game yields our result.

We first update this value function by appropriately choosing the value of k that corresponds to the time $t+1$ to $t + 1/2$. The controller is idle, but the opponent is allowed to chose a randomized policy from a state-dependent compact set. If the state is (i, a), using (54), the set is $\mathscr{P}_i^a$, and the value function is updated as

$$v_{t+1/2}((i, a)) = \max_{p \in \mathscr{P}_i^a} \left(\sum_{j=1}^{n} p(j) v_{t+1}(j) \right), \tag{55}$$

where we make use of the fact that incurred costs are zero in this step. To update the value function from $t+1/2$ to t, we use the fact that the opponent is idle. For $i=1,\ldots,n$, the value function is updated as

$$v_t(i) = \min_{\mathbf{q}\in\Delta_m} \mathbf{E}_{\mathbf{q}}\big(c_t(i,a) + v_{t+1/2}((i,a))\big). \tag{56}$$

The right-hand side of (56) is a linear program in variable $\mathbf{q}$. Thus, the optimal value is obtained at the vertices of the feasible set Δ_m, which correspond to purely deterministic actions. Hence,

$$v_t(i) = \min_{a\in\mathcal{A}}\big(c_t(i,a) + v_{t+1/2}((i,a))\big). \tag{57}$$

Combining (55) and (57) ends our proof.

Appendix B. Properties of Function σ of §5.3

Here, we prove two properties of the function σ involved in the bisection algorithm of §5.3. For simplicity of notation, we assume that there is an unique index i^* achieving the maximum in $v_{\max}$, that is, $v(i^*)=v_{\max}$.

We first show that $\sigma(\mu)\to v_{\max}$ as $\mu\to v_{\max}$. We have

$$\lambda(\mu) = \frac{\mu - v(i^*)}{f(i^*)} + o(\mu - v(i^*)).$$

We then express $\sigma(\mu)$ as

$$\sigma(\mu) = \mu - \lambda(\mu)\Bigg(1+\beta-\beta_{\max}+\log\lambda(\mu) - \sum_{j\neq i^*} f_j \log(\mu - v_j)\Bigg) - \lambda(\mu) f(i^*)\log(\mu - v(i^*)).$$

The second term (first line) vanishes as $\mu\to v_{\max}$, because $\lambda(\mu)\to 0$. In view of the expression of $\lambda(\mu)$ above, the last term (second line) behaves as $(\mu - v(i^*))\log(\mu - v(i^*))$, which also vanishes.

Next, we prove that $\sigma'(\mu)\to-\infty$ as $\mu\to v_{\max}$. We obtain easily

$$\frac{d\lambda(\mu)}{d\mu} = \frac{\sum_j (f(j)/(\mu - v(j))^2)}{\big(\sum_j \big(f(j)/(\mu - v(j))\big)\big)^2} \to \frac{1}{f(i^*)} \quad \text{when } \mu\to v(i^*).$$

We then have

$$\begin{aligned}\frac{\partial h}{\partial\lambda}(\lambda(\mu),\mu) &= \sum_j \log\frac{\lambda(\mu)f(j)}{\mu - v(j)} - \beta \\ &= \log\frac{\lambda(\mu)f(i^*)}{\mu - v(i^*)} + \sum_{j\neq i^*}\log\frac{\lambda(\mu)f(j)}{\mu - v(j)} - \beta \\ &= \log(1+o(1)) + (n-1)\log\lambda(\mu) \\ &\quad + \sum_{j\neq i^*}\log\frac{f(j)}{\mu - v(j)} - \beta \\ &\to -\infty \quad \text{as } \mu\to v(i^*).\end{aligned}$$

Also, by definition of $\lambda(\mu)$, we have $\partial h/\partial\mu(\lambda(\mu),\mu)=0$. The proof is achieved with the identity (44).

Appendix C. Properties of Function σ of §6.3

In this section, we prove that the function σ defined in (47) obeys properties (48), (49), and (50).

First, we prove (49). If $v(j)=v_{\max}$ for every j, the result holds, with $Q(v)=Q(v_{\max}\mathbf{1})=1$. Assume now that there exists j such that $v(j)<v_{\max}$. We have

$$\begin{aligned}\sigma(\lambda) &= \lambda\log\Bigg(e^{v_{\max}/\lambda}\sum_j q(j)\exp\Bigg(\frac{v(j)-v_{\max}}{\lambda}\Bigg)\Bigg) + \beta\lambda \\ &= v_{\max} + \beta\lambda + \lambda\log\Bigg(\sum_{j:v(j)=v_{\max}} q(j) + \sum_{j:v(j)<v_{\max}} q(j)\exp\Bigg(\frac{v(j)-v_{\max}}{\lambda}\Bigg)\Bigg) \\ &= v_{\max} + \beta\lambda + \lambda\log(Q + O(e^{-t/\lambda})) \\ &= v_{\max} + (\beta + \log Q)\lambda + O(\lambda e^{-t/\lambda}),\end{aligned}$$

where $t = v_{\max} - v_s > 0$, where v_s is the largest $v(j) < v_{\max}$. This proves (49).

From the expression of σ given in the second line above, we immediately obtain the upper bound in (48).

The expansion of σ at infinity provides

$$\begin{aligned}\sigma(\lambda) &= \beta\lambda + \lambda\log\Bigg(\sum_j q(j)\Bigg(1 + \frac{v(j)}{\lambda} + o(\lambda)\Bigg)\Bigg) \\ &= q^T v + \beta\lambda + o(1),\end{aligned}$$

which proves (50). The lower bound in (48) is a direct consequence of the concavity of the log function.

Appendix D. Calculation of β for a Desired Confidence Level

In this section, we describe a one-to-one correspondence between a lower bound on the log-likelihood function β, as used in §5, and a desired level of confidence $(1-U_L)$ on the transition matrix estimates, as used in §8. This correspondence is valid for asymptotically large samples only but can serve as a guideline to choose β. The following material is standard; see, for instance, Lehmann (1986).

First, we define a vector $\theta\in\mathbf{R}^{n(n-1)}$ that contains the first $n-1$ columns to be estimated in a $n\times n$ transition matrix P. We order θ so that $P(i,j)=\theta((n-1)(i-1)+j)$ for $1\leqslant i\leqslant n$, $1\leqslant j\leqslant(n-1)$. Using the conditions $P\mathbf{1}=\mathbf{1}$, we can write P as an (affine) function of θ, and express the log-likelihood function $L(P)$ of (37) as a function $l(\theta)$. Let $\hat{\theta}$ be the vector corresponding to the matrix of empirical frequencies F, that we assumed to be positive componentwise. Provided some regularity conditions hold, one can show that for asymptotically large samples, θ is normally distributed with mean given by $\hat{\theta}$, and inverse covariance matrix $H=-E_\theta((\nabla^2 l)(\theta))$. Furthermore, we can approximate H by the observed information matrix $\widehat{H}:=-(\nabla^2 l)(\hat{\theta})$. In our case, the nonzero elements of this

matrix are

$$\widehat{H}((n-1)(i-1)+j,(n-1)(i-1)+k) = \begin{cases} \dfrac{1}{F(i,n)}+\dfrac{1}{F(i,j)} & \text{if } j=k, \\ \dfrac{1}{F(i,n)} & \text{otherwise.} \end{cases}$$

If q denotes the quadratic approximation to l around $\hat{\theta}$, we have

$$q(\theta)=\beta_{\max}-\tfrac{1}{2}(\theta-\hat{\theta})^T\widehat{H}(\theta-\hat{\theta}),$$

where $\beta_{\max}$ is the maximal log-likelihood defined in §5.1. Then, the parameter β is chosen to be the smallest such that, under the Gaussian probability distribution $\mathcal{N}(\hat{\theta},\widehat{H}^{-1})$, the set $\{\theta\colon q(\theta)\geqslant\beta\}$ has probability larger than a given threshold $(1-U_L)$, where (say) $U_L=15\%$ to obtain the 85% confidence level. It turns out that we can solve for such a β explicitly:

$$(1-U_L)=F_{\chi^2_{n(n-1)}}(2(\beta_{\max}-\beta)), \tag{58}$$

where $F_{\chi^2_d}$ is the cumulative density function of the χ^2-distribution with d degrees of freedom. The latter can be approximated as follows (Pitman 1993):

$$F_{\chi^2_d}(\xi)\approx\Phi(z)-\frac{\sqrt{2}}{3\sqrt{d}}(z^2-1)\phi(z), \tag{59}$$

where $z=(\xi-d)/\sqrt{d}$, $\phi(z)=(1/\sqrt{2\pi})e^{-(1/2)z^2}$, and $\Phi(z)=\int_{-\infty}^{z}\phi(u)\,du$ is the standard normal cumulative density function.

Acknowledgments

The authors thank Antar Bandyopadhyay, Bob Barmish, Giuseppe Calafiore, Vu Duong, Mikael Johansson, Yann Le Tallec, Rupak Majumdar, Andrew Ng, Stuart Russell, Shankar Sastry, Ben Van Roy, Michael Todd, and Pravin Varaiya for interesting discussions and comments. The authors are grateful to Dimitris Bertsimas for pointing out a mistake in an earlier version of the paper, and to Alain Haurie for his very detailed comments. They are especially thankful to the unknown reviewers whose interesting comments prompted a significant portion of this work. This research was funded in part by Eurocontrol-014692, DARPA-F33615-01-C-3150, and NSF-ECS-9983874.

References

Abbad, M., J. A. Filar. 1992. Perturbation and stability theory for Markov control problems. *IEEE Trans. Automatic Control* **37** 1415–1420.

Abbad, M., J. Filar, T. Bielecki. 1992. Algorithms for singularly perturbed limiting average Markov control problems. *IEEE Trans. Automatic Control* **37** 1421–1425.

Altman, E., A. Hordijk. 1994. Zero-sum Markov games and worst-case optimal control of queueing systems. *QUESTA* **21**(Special Issue on Optimization of Queueing Systems) 415–447.

Altman, E., E. A. Feinberg, A. Shwartz. 2000. Weighted discounted stochastic games with perfect information. *Ann. Internat. Soc. Dynamic Games* **5** 303–323.

Bagnell, J., A. Ng, J. Schneider. 2001. Solving uncertain Markov decision problems. Technical report CMU-RI-TR-01-25, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Bertsekas, D., J. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Nashua, NH.

Boyd, S., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, Cambridge, UK.

Epstein, L. G., M. Schneider. 2002. Learning under ambiguity. http://www.econ.rochster.edu/Faculty/Epstein.html.

Feinberg, E., A. Shwartz. 2002. *Handbook of Markov Decision Processes, Methods and Applications*. Kluwer Academic Publishers, Boston, MA.

Ferguson, T. 1974. Prior distributions on space of probability measures. *Ann. Statist.* **2**(4) 615–629.

Givan, R., S. Leach, T. Dean. 1997. Bounded parameter Markov decision processes. *Fourth European Conf. Planning*, 234–246.

Iyengar, G. 2003. Robust dynamic programming. Technical report TR-2002-07, Columbia University, New York.

Kalyanasundaram, S., E. Chong, N. Shroff. 2001. Markov decision processes with uncertain transition rates: Sensitivity and robust control. Technical report, Department of ECE, Purdue University, West Lafayette, IN.

Lehmann, E. 1986. *Testing Statistical Hypothesis*. Wiley, New York.

Lehmann, E., G. Casella. 1998. *Theory of Point Estimation*. Springer-Verlag, New York.

Mine, H., S. Osaki. 1970. *Markov Decision Processes*. American Elsevier Publishing, New York.

Nilim, A., L. El Ghaoui. 2002. Robust solution to the Markov decision processes with uncertain transition matrices. Technical report UCB/ERL M02/31, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA.

Nilim, A., L. El Ghaoui, M. Hansen, V. Duong. 2001. Trajectory-based air traffic management (TB-ATM) under weather uncertainty. *Proc. 4th USA/EUROPE ATM R&D Seminar*, Santa Fe, NM, 64–72.

Nowak, A. S. 1984. On zero sum stochastic games with general state space. I. *Probab. Math. Statist.* **4**(1) 13–32.

Pitman, J. 1993. *Probability*. Springer-Verlag, New York.

Poor, H. 1988. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, New York.

Putterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, New York.

Satia, J. K., R. L. Lave. 1973. Markov decision processes with uncertain transition probabilities. *Oper. Res.* **21**(3) 728–740.

Shapiro, A., A. J. Kleywegt. 2002. Minimax analysis of stochastic problems. *Optim. Methods Software*. **17**(1) 523–592.

Siouris, G. 1995. *Optimal Control and Estimation Theory*. Wiley-Interscience, New York.

White, C. C., H. K. Eldeib. 1994. Markov decision processes with imprecise transition probabilities. *Oper. Res.* **42**(4) 739–749.

Wilks, S. 1962. *Mathematical Statistics*. Wiley-Interscience, New York.