ROBUST SOLUTIONS TO MARKOV DECISION PROBLEMS WITH UNCERTAIN TRANSITION MATRICES

by

Laurent El Ghaoui and Arnab Nilim

Memorandum No. UCB/ERL M02/31

5 November 2002

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Robust Solutions to Markov Decision Problems with Uncertain Transition Matrices

Laurent El Ghaoui, elghaoui@eecs.berkeley.edu
Arnab Nilim, nilim@eecs.berkeley.edu (Corresponding Author)
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720, USA

Keywords : Dynamic programming: Markov, finite state, programming: convex, uncertainty, robustness, statistics: estimation.

Abstract

Optimal solutions to Markov Decision Problems (MDPs) may be very sensitive with respect to the state transition probabilities. In many practical problems, the estimation of those probabilities is far from accurate. Hence, estimation errors are, together with the curse of dimensionality, a limiting factor in applying MDPs to real-world problems. We propose an algorithm for solving finite-state and finite-action MDPs, where the solution is guaranteed to be robust with respect to estimation errors on the state transition probabilities. Our algorithm involves a statistically accurate yet numerically efficient representation of uncertainty via likelihood or entropy functions. The worst-case complexity of the robust algorithm is the same as the original Bellman recursion. Hence, robustness can be added at practically no extra computing cost.

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 recourse corresponding to this information during the multi-stage 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 matrix itself is uncertain, and seek a robust decision for it. Our work is motivated by the fact that in most practical problems, the transition matrix has to be estimated from data, which is often a difficult task, see for example (Kalyanasundaram et al., 2001; Shapiro and Kleywegt, 2002; White and Eldeib, 1994), as well as (Satia and Lave, 1973; Givan et al., 1997; Bagnell et al., 2001; Feinberg and Shwartz, 2002; Abbad and Filar, 1992; 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.)

A large part of the recent research effort in the MDP area addresses the "curse of dimensionality" stemming from the exponential growth of problem complexity with size (number of states). Recent references on approximate dynamic programming include (Farias and Roy, 2002) and (Ng and Jordan, 2000). There are however a number of applications for which "exact" solutions of the problem are feasible (Nilim et al., 2001), and the Bellman recursion remains practical. Hence, it is of interest to develop robust solutions to MDPs, based on exact formulations.

A number of authors have addressed the issue of uncertainty in the transition matrix. 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: see (Satia and Lave, 1973; White and Eldeib, 1994; Givan et al., 1997; Bagnell et al., 2001). Although our approach allows to describe the uncertainty on the transition matrix by a polytope, we will argue *against* choosing such a model for the uncertainty. First, a 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; Bagnell et al., 2001). Perhaps more importantly, polytopic models, especially interval matrices, are very poor representations of statistical uncertainty and lead to very conservative robust policies.

We propose here an uncertainty model which results in an algorithm that is *both* statistically accurate and numerically tractable. We develop a formulation in which the concern for robustness can be handled at virtually no additional computational cost. This means that the method is directly applicable to those problems already amenable to exact dynamic programming via Bellman recursions.

Our paper is organized as follows. The problem is set up in section 2. In sections 3, we

describe the so-called likelihood model and some variations. Section 4 examines the entropy models, while section 5 deals with ellipsoidal and "interval matrix" models. Our results are summarized in section 6. We describe numerical results in the context of aircraft routing in section 7.

Notation

$P > 0$ or $P \geq 0$ refers to the strict or non-strict componentwise inequality for matrices or vectors. For a vector p, $\log p$ refers to the componentwise operation.

2 Problem Setup

2.1 The robust Bellman recursion

We consider a Markov decision process with finite state $\mathcal{X}$, finite action set $\mathcal{A}$, with $|\mathcal{X}| = n$ and $|\mathcal{A}| = m$. We denote by $P = (P^a)_{a \in \mathcal{A}}$ the collection of transition matrices, and by $c_t(i, a)$ the cost corresponding to state i and action a at time t, and denote by c_T the cost function at the terminal time, T.

Our *nominal* problem is to minimize the expected cost over a finite horizon:

$$\min_{\pi \in \Pi} \mathbf{E} \left(\sum_{t=0}^{T-1} c_t(i_t, a_t) + c_T(i_T) \right)$$

where $\pi = (a_0, \ldots, a_{T-1})$ denotes the strategy and Π the strategy space. When the transition matrices are exactly known, the value function can be computed via the Bellman recursion

$$V_t(i) = \min_{a \in \mathcal{A}} \left(c_t(i, a) + \sum_{j=1}^{n} P^a(i, j) V_{t+1}(j) \right). \tag{1}$$

Each step of the Bellman recursion has worst-case complexity $O(nm)$.

Now consider the case when the collection of transition matrices, $P = (P^a)_{a \in \mathcal{A}}$, is only known to lie in some given convex subset $\mathcal{P}$ of $\mathcal{T}^m$, where $\mathcal{T}$ is the set of $n \times n$ transition matrices (componentwise non-negative matrices with rows summing to one). Note that for now, we may include in our uncertainty model dependencies between errors in P^a and $P^{a'}$ for different actions a and a'. For a given action a, and state i, we denote by p_i^a the next state distribution drawn from P^a corresponding to state i; thus p_i^a is the i-th row of matrix P^a. We denote by $\mathcal{P}_i^a$ the projection of the set $\mathcal{P}$ onto the set of p_i^a-variables.

We address a robust dynamic programming problem in which the uncertainty on the transition matrices acts as an opponent:

$$\max_{P \in \mathcal{P}} \min_{\pi \in \Pi} \mathbf{E} \left(\sum_{t=0}^{T-1} c_t(i_t, a_t) + r_T(i_T) \right).$$

The corresponding game can be solved via the "robust counterpart" to the above Bellman recursion:

$$V_t(i) = \max_{P \in \mathcal{P}} \min_{a \in \mathcal{A}} \left(c_t(i,a) + \sum_{j=1}^{n} P_{ij}^a V_{t+1}(j) \right). \tag{2}$$

It turns out we can exchange the "min" and "max" operators in the above, as expressed by the following theorem, which is proved in Appendix A.

Theorem 1 *The robust Bellman recursion (2) is equivalent to the following recursion*

$$\begin{aligned} V_t(i) &= \min_{a \in \mathcal{A}} \max_{P_i^a \in \mathcal{P}_i^a} \left(c_t(i,a) + \sum_{j=1}^{n} P_{ij}^a V_{t+1}(j) \right) \\ &= \min_{a \in \mathcal{A}} c_t(i,a) + \phi_{\mathcal{P}_i^a}(V_{t+1}), \end{aligned} \tag{3}$$

where $\phi_{\mathcal{P}_i^a}$ denotes the support function of the (convex) set $\mathcal{P}_i^a$.

One step of the robust Bellman recursion thus involves the solution of a convex optimization problem. Obviously, the complexity of the robust Bellman recursion depends solely on the complexity of the projections $\mathcal{P}_i^a$ for each i and a. Obviously, the set $\mathcal{P}$ should be an accurate (non-conservative) description of statistical uncertainty on the whole collection of transition matrices.

Note that the effect of uncertainty on a *given* strategy $\pi = (a_0, \ldots, a_{T-1})$ can be evaluated by the following recursion

$$V_t(i) = \max_{P_i^a \in \mathcal{P}_i^a} \left(c_t(i,a) + \sum_{j=1}^{n} P_{ij}^a V_{t+1}(j) \right), \tag{4}$$

which provides the worst-case value function for a given strategy.

2.2 Main result

In this paper, we address the problem of efficiently computing the value function via the above recursion. Once the uncertainty model is chosen, the challenge is to solve the "inner problem" in (3), which reduces to computing values of the support function of a given convex set $\mathcal{U}$:

$$\phi_{\mathcal{U}}(v) = \max_{p \in \mathcal{U}} v^T p, \tag{5}$$

where the variable p corresponds to a particular row of a specific transition matrix, $\mathcal{U}$ is the set that describes the uncertainty on this row, and v is an appropriately defined vector with non-negative components, containing the elements of the value function. We refer to the above problem as the *inner problem.*

We will consider various representations of uncertainty. All our models involve *independent* descriptions of the uncertainty on each transition matrix; in other words, we postulate

that $\mathcal{P}$ is a direct product $\bigotimes_{a\in\mathcal{A}} \mathcal{P}^a$, where $\mathcal{P}^a$ describes uncertainty on the transition matrix P^a. This assumption is not formally needed, but simplifies the task of forming the projections $\mathcal{P}_i^a$ required in the robust Bellman recursion (3).

Our main uncertainty model is based on a log-likelihood constraint on each transition matrix. This representation enables one to solve for one step the robust dynamic programming recursion in wort-case time of $O(n\log(1/\epsilon))$ via a simple bisection algorithm, where n is the size of the state space, and ϵ a convergence parameter. This brings the total complexity of one step of the Bellman recursion to $O(nm\log(1/\epsilon))$, where m is the cardinality of the action set. At the same time, our model allows an accurate description of statistical uncertainty on the transition matrix. Hence, non-conservative robustness is obtained at a moderate increase ($\log(1/\epsilon)$) with respect to the classical Bellman recursion. We also describe models based on relative entropy bounds, and obtain similar results.

We will also consider perhaps more classical ways to describe uncertainty, among which an interval models based on componentwise intervals of confidence, and ellipsoidal models that are based on quadratic approximations to the log-likelihood. We will observe that some of these descriptions give rise to similar low complexity results. However, these "approximate" models, it may be argued, are less statistically accurate.

3 Likelihood Models

Our first model is based on a likelihood constraint to describe uncertainty on each transition matrix. We denote by F^a the matrix of empirical frequencies of transition with control a observed over a given time interval; denote by f_i^a its i^{th} row. We have $F^a \geq 0$ and $F^a 1 = 1$, where 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 i^{th} 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 $\mathcal{P}^a$ and $\mathcal{P}_i^a$. When the meaning is clear from context, we will further drop the subscript i.

3.1 Model description

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

$$\max\ L(P) := \sum_{i,j} F(i,j)\log P(i,j)\ :\ P \geq 0,\ \ P1 = 1$$

The optimal log-likelihood is $\beta_{\max} = \sum_{i,j} F(i,j)\log F(i,j)$.

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

$$\left\{ P \in \mathbf{R}^{n\times n}\ :\ P \geq 0,\ \ P1 = 1,\ \sum_{i,j} F(i,j)\log P(i,j) \geq \beta \right\}, \tag{6}$$

where $\beta < \beta_{\max}$ is given number. In practice, β can be estimated using resampling methods, or a large-sample Gaussian approximation, so as to ensure that the set above achieves a given level of confidence (see Appendix D).The above description is classical in the sense that it is the starting point for the estimation of statistical ellipsoids of confidence; see section 5.2 for further details.

In our problem, we only need to work with the uncertainty on each row p_i, that is, with *projections* of the set above. Due to the separable nature of the maximum-likelihood problem, the projection of the above set onto the p_i variables of matrix P can be given explicitly, as

$$\mathcal{P}_i(\beta) := \left\{ p \in \mathbf{R}^n \ : \ p \geq 0, \ \ p^T \mathbf{1} = 1, \ \ \sum_j f_i(j) \log p_i(j) \geq \beta_i \right\},$$

where

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

3.2 The dual problem

We are now ready to attack problem (5) under the premise that the transition matrix is only known to lie in some likelihood region as defined above. The inner problem is to compute

$$\phi := \max_p \ p^T v \ : \ p \geq 0, \ \ p^T \mathbf{1} = 1, \ \ \sum_j f(j) \log p(j) \geq \beta,$$

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 non-empty interior.

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

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

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

$$d(\nu, \mu, \lambda) = \sup_p \mathcal{L}(v, \nu, \mu, \lambda) = \sup_p (p^T v + \nu^T p + \mu(1 - p^T \mathbf{1}) + \lambda(f^T \log p - \beta)). \qquad (7)$$

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

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

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

$$\overline{\phi} := \min_{\lambda, \mu, \nu} \ \mu - (1 + \beta)\lambda + \lambda \sum_j f(j) \log \frac{\lambda f(j)}{\mu - v(j) - \nu(j)} \ : \ \lambda \geq 0, \ \ \nu \geq 0, \ \ \nu + v \leq \mu \mathbf{1}.$$

Since the above problem is convex, and has a feasible set with non-empty interior, there is no duality gap, that is, $\phi = \overline{\phi}$. Moreover, by a monotonicity argument, we obtain that the optimal dual variable ν is zero, which reduces the number of variables to two:

$$\phi = \min_{\lambda,\mu} h(\lambda, \mu)$$

where

$$h(\lambda,\mu) := \begin{cases} \mu - (1+\beta)\lambda + \lambda \sum_j f(j) \log \dfrac{\lambda f(j)}{\mu - v(j)} & \text{if } \lambda > 0, \ \ \mu > v_{\max} := \max_j v(j), \\ +\infty & \text{otherwise.} \end{cases} \tag{8}$$

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

$$\nabla h(\lambda,\mu) = \begin{bmatrix} \sum_j f(j) \log \dfrac{\lambda f(j)}{\mu - v(j)} - \beta \\ 1 - \lambda \sum_j \dfrac{f(j)}{\mu - v(j)} \end{bmatrix}. \tag{9}$$

3.3 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

$$\lambda(\mu) = \left(\sum_j \frac{f(j)}{\mu - v(j)} \right)^{-1}, \tag{10}$$

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

$$\phi = \min_{\mu \geq v_{\max}} \phi(\mu),$$

where $v_{\max} = \max_j v(j)$, and $\phi(\mu) = h(\lambda(\mu), \mu)$. We define $\bar{v} = f^T v$, which is the average of v under f. Since h is jointly convex in both its arguments, the function ϕ is convex on its domain ($v_{\max} + \infty$); hence we may use bisection to minimize ϕ.

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

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

$$\phi'(\mu) = \frac{\partial h}{\partial \mu}(\lambda(\mu), \mu) + \frac{\partial h}{\partial \lambda}(\lambda(\mu), \mu) \frac{d\lambda(\mu)}{d\mu}$$

The second 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

$$\frac{\partial h}{\partial \lambda}(\lambda(\mu), \mu) = \sum_j f(j) \log \frac{\lambda(\mu) f(j)}{\mu - v(j)} - \beta > 0. \tag{11}$$

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

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

The above, combined with the bound on $\lambda(\mu)$: $\lambda(\mu) \geq \mu - v_{\max}$, yields a sufficient condition for (11) to hold:

$$\mu > \mu_+^0 := \frac{v_{\max} - e^{\beta - \beta_{\max}} \bar{v}}{1 - e^{\beta - \beta_{\max}}}. \tag{12}$$

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

The bisection algorithm goes as follows:

1. Set $\mu_- = v_{\max}$ and $\mu_+ = \mu_+^0$ as in (12). Let $\epsilon > 0$ be a small convergence parameter.
2. While $\mu_+ - \mu_- > \epsilon(1 + \mu_- + \mu_-)$, repeat
   (a) Set $\mu = (\mu_+ + \mu_-)/2$.
   (b) Compute the gradient of ϕ at μ.
   (c) If $\phi'(\mu) > 0$, set $\mu_+ = \mu$; otherwise, set $\mu_- = \mu$.
   (d) go to 2a.

Each iteration of the above algorithm has worst-case complexity of $O(n)$. The number of iterations grows as $\log(\mu_+^0 - v_{\max})/\epsilon)$, which is independent of problem size. Hence, the worst-case complexity of the algorithm is $O(n)$. Therefore, the cost of adding robustness under the likelihood uncertainty model is $O(1)$, which means that robustness can be added at practically no extra cost.

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. Since we are only interested in the value of the minimum (and not of the minimizer), we may modify the stopping criterion to

$$\mu_+ - \mu_- \leq \epsilon(1 + \mu_- + \mu_-) \text{ or } \phi'(\mu_+) - \phi'(\mu_-) \leq \epsilon.$$

This second criterion retains the same complexity as the original bisection algorithm. The second condition in the criterion implies that $|\phi'((\mu_+ + \mu_-)/2)| \leq \epsilon$, which is an approximate condition for global optimality.

3.4 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 $\{p \ : \ p \geq 0, \ \ p^T 1 = 1\}$. 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 $\alpha \geq \mathbf{1}$ is given, and K is a normalizing constant. Choosing $\alpha = \mathbf{1}$ we recover the "non-informative prior", which is the uniform distribution on the n-dimensional simplex.

The resulting MAP estimation problem takes the form

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

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

$$\phi := \max_p \ p^T v \ \ : \ \ p \geq 0, \ \ p^T \mathbf{1} = 1, \ \ \sum_j (f(j) + \alpha(j) - 1) \log p(j) \geq \gamma,$$

where γ 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 likelihood function, provided we replace f by $f + \alpha - \mathbf{1}$. Therefore, the same results apply to the MAP case.

4 Entropy Models

4.1 Model description

Here, we describe the uncertainty on each row of the transition matrix via an entropy constraint. Specifically we consider problem (5), with the uncertainty on the i-th row of the

transition matrix P^a described via a lower bound on the entropy function relative to a given distribution q (Kullback-Leibler divergence)

$$\mathcal{U}(\beta) = \left\{ p \in \mathbf{R}^n \ : \ p^T 1 = 1, \ \sum_j p(j) \log \frac{p(j)}{q(j)} \leq \beta \right\}.$$

Here, $q > 0$ is a given distribution, and $\beta > 0$ is fixed. Together with $q > 0$, the condition $\beta > 0$ ensures that $\mathcal{U}$ has non-empty interior. (As before, we have dropped the control and row indices a and i).

We now address the inner problem (5), with $\mathcal{U} = \mathcal{U}(\beta)$ given above. We note that the above set actually equals the whole probability simplex if β is too large, specifically if $\beta \geq \max_i(-\log q_i)$, since the latter quantity is the maximum of the relative entropy function over the simplex. Thus, if $\beta \geq \max_i(-\log q_i)$, the worst-case value of $p^T v$ for $p \in \mathcal{U}(\beta)$ is equal to $v_{\max}$.

4.2 Dual problem

By standard duality arguments (set $\mathcal{U}$ being strictly feasible), 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

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

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

$$\min_{\lambda>0} \ \phi(\lambda)$$

where ϕ is the convex function:

$$\phi(\lambda) = \lambda \log \left(\sum_j q(j) \exp \frac{v(j)}{\lambda} \right) + \beta\lambda. \tag{13}$$

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.

4.3 A bisection algorithm

As proved in Appendix C, the convex function ϕ in (13) has the following properties:

$$\forall\, \lambda \geq 0,\ \ q^T v + \beta\lambda \leq \phi(\lambda) \leq v_{\max} + \beta\lambda, \tag{14}$$

and

$$\phi(\lambda) = v_{\max} + (\beta + \log Q(v))\lambda + o(\lambda), \tag{15}$$

where

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

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

$$\phi(\lambda) = q^T v + \beta\lambda + o(1). \tag{16}$$

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

Note that if $\phi'(0) \geq 0$, then $\lambda = 0$ is optimal and the optimal value of ϕ 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 4.1 a similar phenomenon brought about by too large values of β, which resulted in a set $\mathcal{U}$ equal to the probability simplex. Here, the limiting value $-\log Q(v)$ depends not only on q but also on v, since we are dealing with the optimization problem (5) and not only with its feasible set $\mathcal{U}$.

5 Other Specific Models

5.1 Interval matrix model

The *interval matrix* model is when

$$\mathcal{U} = \left\{p\ :\ \underline{p} \leq p \leq \overline{p},\ \ p^T\mathbf{1} = 1\right\},$$

where $p_\pm$ are given componentwise non-negative n-vectors (whose elements do not necessarily sum to one), with $p_+ \geq p_-$. 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 section 5.2). In what follows, we assume that $\mathcal{U}$ is not empty.

Since the inner problem

$$\phi := \max_p\ v^T p\ :\ p \geq 0,\ \ p^T\mathbf{1} = 1,\ \ \underline{p} \leq p \leq \overline{p}$$

is a linear, feasible program, it is equivalent to its Lagrange dual, which has the form

$$\phi = \min_{\mu} (\overline{p} - \underline{p})^T (\mu \mathbf{1} - v)^+ + v^T \overline{p} + \mu(1 - \overline{p}^T \mathbf{1}),$$

where z^+ stands for the positive part of vector z. The function to be minimized is a convex piecewise linear function with break points $v_0 = 0$, $v_1, \ldots, v_n$. Since the original problem is feasible, we have $\mathbf{1}^T \underline{p} \leq 1$, 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 $O(n^2)$, since 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 $O(n \log n)$.

5.2 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 (6):

$$\mathcal{P}(\beta) = \left\{ P \in \mathbf{R}^{n \times n} \ : \ P \geq 0, \ \ P\mathbf{1} = \mathbf{1}, \ \ Q(P) \geq \beta \right\}, \tag{17}$$

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, the projection on the space of i^{th} row variables assumes a similar shape, that of an ellipsoid intersected with the probability simplex, specifically

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

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

$$\max_{p} \ v^T p \ : \ p \geq 0, \ \ p^T \mathbf{1} = 1, \ \ \sum \frac{(p_i(j) - f_i(j))^2}{f_i(j)} \leq \kappa^2.$$

According to (Nesterov and Nemirovski, 1994), the above problem has worst-case complexity of $O(n^{3.5})$. This brings the complexity of one step of the robust Bellman recursion to $O(n^{3.5} m)$.

In statistics, it is a standard practice to further simplify the description above, by relaxing the inequality constraints $P \geq 0$ in the definition of $\mathcal{P}(\beta)$. We thus obtain the (unconstrained) *ellipsoidal* model, which leads to

$$\phi := \max_p \; v^T p \; : \; p^T \mathbf{1} = 1, \; \sum \frac{(p_i(j) - f_i(j))^2}{f_i(j)} \leq \kappa^2.$$

Taking the dual of the above problem, we obtain the closed-form expression

$$\phi = f_i^T v + \kappa \sqrt{\sum_j f_i(j)(v(j) - f_i^T v)^2},$$

which has $O(n)$ complexity. The robust recursion based on the unconstrained ellipsoidal model is thus $O(nm)$, the same as that of the classical Bellman recursion.

This economical computation comes at an expense, which is the possible conservatism of the worst-case value function stemming from our neglect of the non-negativity constraints on the transition matrix. Another potential problem is the fact that the ellipsoid model is symmetric around the maximum-likelihood point, which might not be realistic . In the maximum-likelihood model, the non-negativity constraints are implicit in the likelihood bound, and the model yields potentially non-symmetric (hence more realistic) estimates.

Uncertainty on the reference distribution q in entropy models. We may generalize the relative entropy models to the case when there is uncertainty on the reference distribution q.

If $\mathcal{Q}$ is a set of reference distributions q, we can consider the inner problem (5), where the uncertainty set $\mathcal{U}$ replaced by one of the form

$$\mathcal{U} = \left\{ p \; : \; p \geq 0, \; p^T \mathbf{1} = 1, \; \sum_j p(j) \log \frac{p(j)}{q(j)} \leq \beta \text{ for some } q \in \mathcal{Q} \right\}.$$

Using the same steps as before, the inner problem reduces to

$$\max_{q \in \mathcal{Q}} \min_{\lambda > 0} \; \lambda \log \left(\sum_j q(j) \exp \frac{v(j)}{\lambda} \right) + \beta \lambda.$$

The above problem is very easy if $\mathcal{Q}$ is a box (hyperrectangle) or an ellipsoid parallel to the coordinate axes. For example, assume that $\mathcal{Q}$ assumes the form we encountered in the case of ellipsoidal models, specifically $\mathcal{Q} = \mathcal{P}$, where $\mathcal{P}$ is given by (17). Then we obtain

$$\min_{\lambda > 0} \; \lambda \log \left(\sum_j f(j) \exp \frac{v(j)}{\lambda} + \kappa \sqrt{\sum_j f(j) (\exp \frac{v(j)}{\lambda} - f^T \exp \frac{v}{\lambda})^2} \right) + \beta \lambda.$$

A bisection algorithm similar to the ones described earlier can be applied to this modified problem.

6 Robust Algorithm Summary

The robust Dynamic Programming Algorithm is as follows.

1. Initialize the value function to its terminal value V_T.
2. Repeat until $t = 0$:
   (a) For all states i and controls a, compute the solution to the inner problem
   $$\phi(i,a) = c_t(i,a) + \max_{p \in \mathcal{P}_i^a} p^T v_t$$
   (b) Update the value function by
   $$v_{t+1}(i) = \min_{a \in \mathcal{A}} c_t(i,a) + \phi(i,a)$$
   (c) Replace t by $t-1$ and go to 2.

7 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.

7.1 The nominal problem

In (Nilim et al., 2001), we have introduced 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 weather Markov chain 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 airspace 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 evolves 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 airspace 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).

Since 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.

7.2 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 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 on the likelihood function β, which is a measure of the uncertainty level. Its maximum value β_{max} corresponds to the case with no uncertainty, and decreasing values of β correspond to higher uncertainty level. To β, we may associate a measure of uncertainty that is perhaps more readable: the *uncertainty level*, denoted 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.

7.3 Comparing robust and nominal strategies

In figure 3, we compare various strategies: we plot the relative delay, which is the relative increase 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 the recursion (4), 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 on 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 β_{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 β_{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 14% with the robust strategy. In both of 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, the nominal strategy very quickly reaches delay values comparable to those obtained with the conservative strategy, as the uncertainty level increases. 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 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 4 seconds, and the robust version took 4 more seconds to solve.

7.4 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 the 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 .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 in order 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.

8 Concluding remarks

We have considered uncertainty models on the transition matrix that are statistically accurate and give rise very moderate increase in computational cost. All the models, (except the interval matrix model), considered here give rise to inner problems with worst-case complexity less than $O(n)$. With these models, the total cost of one step of the robust Bellman recursion is thus $O(mn)$ (m is the number of actions). This has the same same complexity as the classical recursion, which has complexity of $O(mn)$. In the interval matrix model, the the worst-case complexity is $O(mn \log n)$.

From the point of view of statistical accuracy, the likelihood or entropy models are certainly preferable to the ellipsoid or interval models: these models take into account sign

constraints, possibly asymmetric uncertainty around the maximum-likelihood or minimum relative entropy point, in contrast of the ellipsoidal and box uncertainty models that are possibly crude approximations to the above models.

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 expected flight delay.

Acknowledgments

The authors would like to thank Antar Bandyopadhyay, Ashwin Ganesan, Jianghai Hu, Mikael Johansson, Andrew Ng, Stuart Russell, Shankar Sastry, and Pravin Varaiya for interesting discussions and comments. The authors are specially grateful to Dimitri Bertsimas for pointing out an important mistake in the earlier version of the paper.

This research is funded in part by Eurocontrol-014692, DARPA-F33615-01-C-3150, and NSF-ECS-9983874.

A Proof of Theorem 1

In this section, we show the equivalence between the two recursions (2) and (3). For a given state i, and value function vector v, we consider the problem of computing

$$\psi = \max_{P \in \mathcal{P}} \min_{a \in \mathcal{A}} c(i,a) + v^T p_i^a.$$

Denote by $\mathcal{S}$ the probability simplex in $\mathbf{R}^m$. We have

$$\begin{aligned}
\psi &= \max_{P \in \mathcal{P}} \min_{\lambda \in \mathcal{S}} \sum_a \lambda(a) \left(c(i,a) + v^T p_i^a \right) \\
&= \min_{\lambda \in \mathcal{S}} \max_{P \in \mathcal{P}} \sum_a \lambda(a) \left(c(i,a) + v^T p_i^a \right) \\
&= \min_{\lambda \in \mathcal{S}} \sum_a \lambda(a) \left(c(i,a) + \max_{P \in \mathcal{P}} v^T p_i^a \right) \\
&= \min_{\lambda \in \mathcal{S}} \sum_a \lambda(a) \left(c(i,a) + \max_{P_i^a \in \mathcal{P}_i^a} v^T p_i^a \right) \\
&= \min_{\lambda \in \mathcal{S}} \sum_a \lambda(a) \left(c(i,a) + \phi_{\mathcal{P}_i^a}(v) \right) \\
&= \min_{a \in \mathcal{A}} \left(c(i,a) + \phi_{\mathcal{P}_i^a}(v) \right),
\end{aligned} \tag{18}$$

where the second line follows from standard linear programming duality arguments, with $\mathcal{P}$ and $\mathcal{S}$ compact. This achieves the proof.

B Properties of function ϕ of section 3.3

Here, we prove two properties of the function ϕ involved in the bisection algorithm of section 3.3. 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 $\phi(\mu) \to v_{\max}$ as $\mu \to v_{\max}$. We have

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

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

$$\begin{aligned}\phi(\mu) = \mu - \lambda(\mu) \left(1 + \beta - \beta_{\max} + \log \lambda(\mu) - \sum_{j \neq i^*} f_j \log(\mu - v_j) \right) \\ - \lambda(\mu) f(i^*) \log(\mu - v(i^*)).\end{aligned}$$

The second term (first line) vanishes as $\mu \to v_{\max}$, since $\lambda(\mu) \to 0$ then. 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 $\phi'(\mu) \to -\infty$ as $\mu \to v_{\max}$. We obtain easily

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

We then have

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

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

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

C Properties of function ϕ of section 4.3

In this section, we prove that the function ϕ defined in (13) obeys properties (14), (15) and (16).

First, we prove (15). If $v(j) = v_{\max}$ for every j, the result holds, with $Q(v) = Q(v_{\max}\mathbf{1}) = 1$. Assume now that there exist j such that $v(j) < v_{\max}$. We have

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

where $t = v_{\max} - v_s > 0$, where v_s is the largest $v(j) < v_{\max}$. This proves (15).

From the expression of ϕ given in the second line above, we immediately obtain the upper bound in (14).

The expansion of ϕ at infinity provides

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

which proves (16). The lower bound in (14) is a direct consequence of the concavity of the log function.

D Calculation of β for a Desired Confidence Level

In this section, we describe the one-to-one correspondence between a lower bound on the likelihood function, as used in section 3, with a desired level of confidence $(1 - U_L)$ on the transition matrix estimates. This correspondence is valid for asymptotically large samples only but can serve as a guideline to choose β.

First, we define a vectors $q_i = [P(i,1), \ldots, P(i,n-1)]^T, \forall i = 1, \ldots, n$ and $\theta = [q_1, \ldots, q_n]^T \in \mathbf{R}^{n(n-1)}$, where P is the transition matrix that we want to estimate. Hence, $P(i,j) = \theta_{ij} = \theta((n-1)^2 i + j)\ \forall\ 1 \le i \le n, 1 \le j \le (n-1)$. Provided some regularity conditions hold (Lehmann, 1986), it is possible to make Laplace approximation of the Likelihood function and we can make the following asymptotic statement about the distribution of θ: precisely, that θ is normally distributed with the mean given by $\hat{\theta}_{ij} := F(i,j),\ 1 \le i \le n, 1 \le j \le (n-1)$ and covariance matrix $I(\theta)$ (Fisher Information matrix) given by

$$I(\theta)_{pq} = E_\theta\left(-\frac{\partial^2}{\partial\theta_p \partial\theta_q} l(\theta)\right)\ \forall\ p, q = 1, \ldots, n(n-1), \tag{19}$$

where $l(.) = \log(L(.))$ is the log-likelihood function.

We can approximate $I(\theta)$ with the observed information matrix, which is meaningful in the neighborhood of $\hat{\theta}$. The equation of the observed information matrix is given by

$$I_o(\theta)_{pq} = -\frac{\partial^2}{\partial\theta_p\partial\theta_q} l(\theta) \; \forall \; p, q = 1, \ldots, n(n-1), \tag{20}$$

where $\frac{\partial^2}{\partial\theta_p\partial\theta_q} l(\theta)$ can be shown to be

$$\frac{\partial^2}{\partial\theta_p\partial\theta_q} l(\theta) = \begin{cases} -\frac{F(p,q)+F(q,n)}{F(p,n)F(p,q)}, & \text{if } p, q \text{ correspond to the elements in a same row in } P \text{ and } p = q, \\ -\frac{1}{F(p,n)}, & \text{if } p, q \text{ correspond to the elements in a same row in } P \text{ and } p \neq q, \\ 0, & \text{if } p \text{ and } q \text{ correspond to the elements in different rows in } P. \end{cases} \tag{21}$$

This is true for large number of sample (Lehmann and Casella, 1998). We further define, $H^{-1} := I_o(\theta)$. Then the parameter β is chosen to be the smallest such that, under the probability distribution $N(\hat{\theta}, (H)^{-1})$, the set,

$$\xi_\beta = \{\theta : \tilde{l}(\theta) \geq \beta\}, \tag{22}$$

where $\tilde{l}(\theta)$ is the quadratic approximation to $l(\theta)$ around $\theta = \hat{\theta}$, that is,

$$\tilde{l}(\theta) = \beta_{max} - \frac{1}{2}(\theta - \hat{\theta})^T H(\theta - \hat{\theta}), \tag{23}$$

has the probability larger than a threshold $(1 - U_L)$, where (say) $U_L = 15\%$ in order to obtain the 85% confidence level.

It turns out that, we can solve for such a β explicitly,

$$(1 - U_L) = F_{\chi^2_{n(n-1)}}(2(\beta_{max} - \beta)), \tag{24}$$

where $F_{\chi^2_{n(n-1)}}(.)$ is the cumulative χ^2 distribution with the degrees of freedom $n(n-1)$, which can be approximated by the following equation (Pitman, 1993)

$$F_{\chi^2_{n(n-1)}}(2(\beta_{max} - \beta)) \approx \Phi(z) - \frac{\sqrt{2}}{3\sqrt{n(n-1)}}(z^2 - 1)\phi(z) \approx U_L, \tag{25}$$

where, $z = \frac{2(\beta_{max}-\beta)-n(n-1)}{\sqrt{2n(n-1)}}$, $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$ and $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}p^2} dp$ is the standard normal cumulative density function.

References

Abbad, M. and Filar, J. (1992). Perturbation and stability theory for markov control problems. *IEEE Transactions on Automatic Control*, 37:1415–1420.

Abbad, M., Filar, J., and Bielecki, T. (1992). Algorithms for singularly perturbed limiting average markov control problems. *IEEE Transactions on Automatic Control*, 37:1421–1425.

Bagnell, J., Ng, A., and Schneider, J. (2001). Solving uncertain markov decision problems. Technical Report CMU-RI-TR-01-25, Robotics Institute, Carnegie Mellon University.

Berstsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming.* Athena Scientific, Massachusetts.

Farias, D. D. and Roy, B. V. (2002). The linear programming approach to approximate dynamic programming. submitted to Operations Research.

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

Ferguson, T. (1974). Prior distributions on space of probability measures. *The Annal of Statistics*, 2(4):615–629.

Givan, R., Leach, S., and Dean, T. (1997). Bounded parameter markov decision processes. In *fourth European Conference on Planning*, pages 234–246.

Kalyanasundaram, S., E.Chong, and Shroff, N. (2001). Markov decision processes with uncertain transition rates: Sensitivity and robust control. Technical report, Department of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA.

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

Lehmann, E. and Casella, G. (1998). *Theory of point estimation.* Springer-Verlag, New York, USA.

Mine, H. and Osaki, S. (1970). *Markov Decision Processes.* American Elsevier Publishing Company Inc.

Nesterov, Y. and Nemirovski, A. (1994). *Interior point polynomial methods in convex programming: Theory and applications.* SIAM, Philadelphia, PA.

Ng, A. and Jordan, M. (2000). Pegasus: A policy search method for large mdps and pomdps. In *the proceedings of the Sixteenth Conference in Uncertainty in Artificial Intelligence.*

Nilim, A., Ghaoui, L. E., Hansen, M., and Duong, V. (2001). Trajectory-based air traffic management (tb-atm) under weather uncertainty. In *the proceeding of the 4th USA/EUROPE ATM R & D Seminar.*

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

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-Interscince, New York.

Satia, J. and Lave, R. L. (1973). Markov decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740.

Shapiro, A. and Kleywegt, A. J. (2002). Minimax analysis of stochastic problems. *Optimization Methods and Software.* to appear.

Siouris, G. (1995). *Optimal Control and estimation theory.* Wiley-Interscience, New York, USA.

White, C. C. and Eldeib, H. K. (1994). Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749.

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

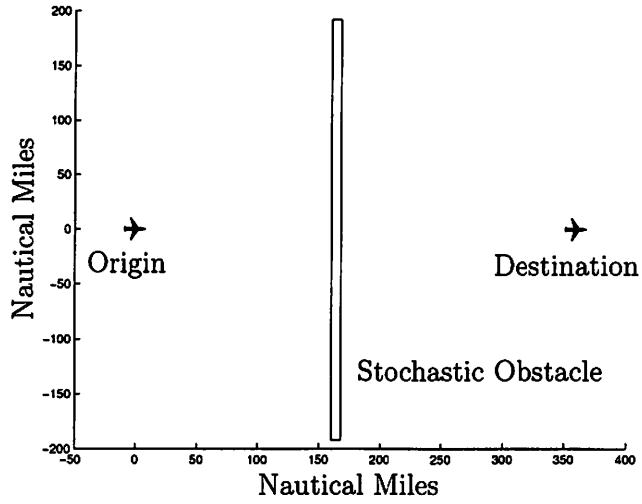

Figure 1: Aircraft Path Planning Scenario.

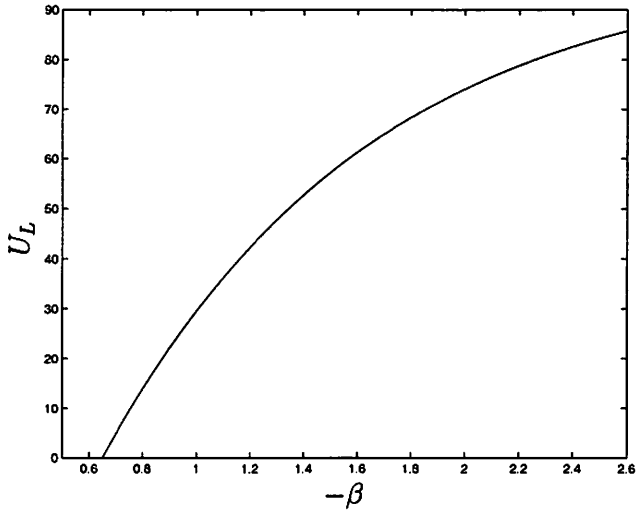

Figure 2: $-\beta$ (negative lower bound on the log-likelihood function) vs U_L (Uncertainty Level (in %) of the Transition Matrices).

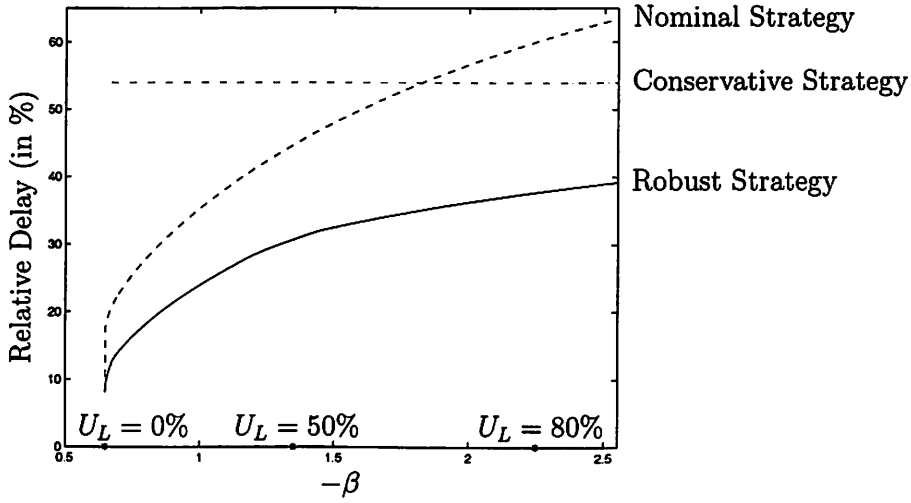

Figure 3: Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function), for both the classical Bellman recursion and its robust counterpart.

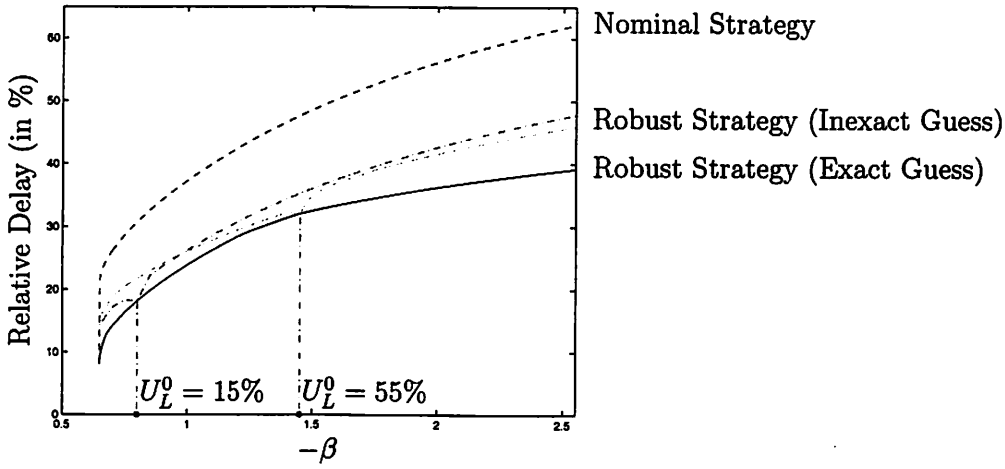

Figure 4: Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function), for the classical Bellman recursion and its robust counterpart (with exact and inexact predictions of the uncertainty level U_L).

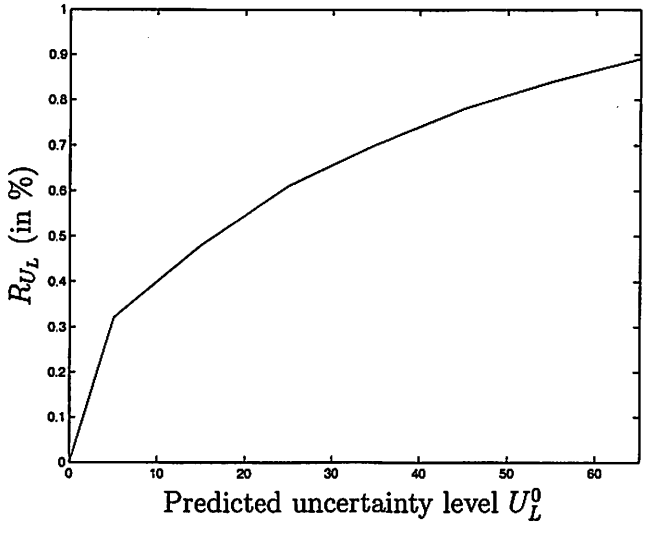

Figure 5: Predicted uncertainty level U_L^0 vs. R_{U_L}, which is the range of the actual uncertainty level U_L over which the nominal strategy performs better than a robust strategy computed with the imperfect prediction U_L^0.