# Piecewise Linear Dynamic Programming for Constrained POMDPs

**Joshua D. Isom**
Sikorsky Aircraft Corporation
Stratford, CT 06615
joshua.isom@sikorsky.com

**Sean P. Meyn** and **Richard D. Braatz**
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{meyn,braatz}@uiuc.edu

## Abstract

We describe an exact dynamic programming update for constrained partially observable Markov decision processes (CPOMDPs). State-of-the-art exact solution of unconstrained POMDPs relies on implicit enumeration of the vectors in the piecewise linear value function, and pruning operations to obtain a minimal representation of the updated value function. In dynamic programming for CPOMDPs, each vector takes two valuations, one with respect to the objective function and another with respect to the constraint function. The dynamic programming update consists of finding, for each belief state, the vector that has the best objective function valuation while still satisfying the constraint function. Whereas the pruning operation in an unconstrained POMDP requires solution of a linear program, the pruning operation for CPOMDPs requires solution of a mixed integer linear program.

## Background

The partially observable Markov decision process (POMDP) is a model for decision-making under uncertainty with respect both to the current state and to the future evolution of the system. The model generalizes the fully observed Markov decision process, which allows only for uncertainty as to the future evolution of the system. The theory of solution of fully observed Markov decision processes with finite or countable state space is well-established for both unconstrained (Puterman 2005) and constrained (Altman 1999) problem formulations. Despite a recent burst of algorithmic development for unconstrained POMDPs (Cassandra 1998; Hansen 1998; Poupart 2005; Feng & Zilberstein 2004; Spaan & Vlassis 2005), there has been relatively little development of algorithmic approaches for the constrained problem.

A comprehensive treatment of *countable state* constrained Markov decision processes is Altman's monograph (Altman 1999). The central solution concept is a countably infinite linear program, which has desirable theoretical properties but obviously requires LP approximation or state truncation to be important practically. Complexity reduction is addressed in (Meyn 2007) for network models. Approaches include relaxation techniques and value function approximations based on a mean-flow model. Because the indefinite-horizon and infinite-horizon discounted POMDPs considered here are equivalent to Markov decision processes with an uncountably infinite state space, the countable state results are not directly applicable.



The solution technique that we propose is related to an approach developed by Piunovskiy and Mao (Piunovskiy & Mao 2000). They considered an uncountably infinite-state Markov decision process, and proposed augmenting the state space with a state representing the expected value of the constraint function, with a penalty function used to prohibit solutions that violate the constraint. The technique of Piunovskiy and Mao is independent of the representation of the value function, and was illustrated with an example that admitted an exact analytical solution. Our approach is similar in that it disallows at every step a solution that would violate the constraint. However, our method uses a piecewise linear representation of the value function, building on techniques developed for unconstrained POMDPs and allowing for $\epsilon$-exact numerical solutions. Our value function update approach can be the basis for value iteration or for policy iteration using Hansen's policy improvement technique.

A constrained partially observable Markov decision process is an 7-tuple

$$(\mathcal{S}, \mathcal{A}, \Sigma, p(s'|s,a), p(z|s), r(s,a), c(s,a)), \qquad (1)$$

where

- $\mathcal{S}$, $\mathcal{A}$ and $\Sigma$ are finite sets of *states, actions,* and *observations,*
- $p(s'|s,a)$ is the *state transition function,* where $p(s'|s,a)$ is the probability that state $s'$ is reached from state $s$ on action $a$,
- $p(z|s)$ is the *observation function,* where $p(z|s)$ is the probability that observation $z \in \Sigma$ will be made in state $s$,
- $r(s,a) \geq 0$ is the *cost* incurred by taking action $a$ in state $s$, and
- $c(s,a) \geq 0$ is a *constraint function*.

A deterministic stationary policy $\phi$ for a CPOMDP is a map from all possible observation sequences to an action in $\mathcal{A}$, with the set of all policies given by

$$\Phi = \{\phi : \Sigma^* \to \mathcal{A}\}. \qquad (2)$$

We consider two problems on CPOMDPs. The indefinite-horizon problem takes the form

$$\text{minimize } E_\Phi \left[ \sum_{t=1}^{T} r(a_t, s_t) \right], \tag{3}$$

subject to

$$E_\Phi \left[ \sum_{t=1}^{T} c(a_t, s_t) \right] \leq \alpha, \qquad \forall s_0, \tag{4}$$

where $T$ is the time that the system enters a set of one or more terminal states, and $s_0$ is the initial state. Conditions that ensure that the dynamic programming operator for an indefinite-horizon problem has a unique fixed point are developed in the literature (Hansen 2007; Patek 2001).

For the infinite-horizon constrained case, the constrained problem takes the form

$$\text{minimize } E_\Phi \left[ \sum_{t=1}^{\infty} \lambda^t r(a_t, s_t) \right] \tag{5}$$

subject to

$$E_\Phi \left[ \sum_{t=1}^{\infty} \lambda^t c(s_t, a_t) \right] \leq \alpha, \qquad \forall s_0, \tag{6}$$

with discount factor $0 < \lambda < 1$ and initial state $s_0$. Discounting ensures that the dynamic programming operator is a contraction operator.

## Motivating Problems

Below we discuss three examples that motivate development of techniques for solution of CPOMDPs. Practical problems that are naturally formulated as CPOMDPs are at least as prevalent as those formulated as unconstrained problems.

### Change Detection

A classic example of a constrained indefinite-horizon POMDP is the problem of quickest change detection. The problem is to minimize detection delay subject to a constraint on the probability of false alarm. This problem was studied by Shiryaev (Shiryaev 1963), who elucidated the form of the optimal policy but did not establish numerical techniques for optimal parameterization of the solution. The difficulty of finding the optimal parameterization for the policy, or even evaluating the probability of false alarm, has been noted in the literature (Tartakovsky & Veeravalli 2004).

An indefinite-horizon constrained Bayes change detection problem is a 5-tuple $\{\Sigma, \alpha, f_0, f_1, \mathcal{T}\}$ with $\Sigma$ a finite set of observations, change time parameter $0 < \rho < 1$, probability of false alarm constraint $0 < \alpha < 1$, pre-change observation probability mass function $f_0 : \Sigma \to [0, 1]$, post-change observation probability mass function $f_1 : \Sigma \to [0, 1]$, and countable time set $\mathcal{T}$. The change time $\nu$ is a geometric random variable with parameter $\rho$, and $\eta$ is an adapted estimate.

The problem statement is

$$\underset{\phi \in \Phi}{\text{minimize}} \underset{\phi(y_1, \ldots, y_h)}{E} [(\eta - \nu - 1)^+] \tag{7}$$

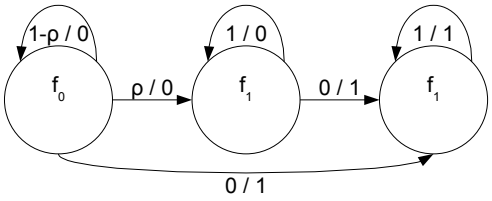


Figure 1: State transition diagram for the change detection problem with geometric change time distribution. Arcs are labeled with transition probabilities under the no-alarm / alarm actions. Nodes are labeled with the distribution of observations for the state.

subject to

$$p(\eta < \nu) < \alpha \tag{8}$$

with

$$\Phi = \{\phi : \Sigma^* \to [0, 1]\}, \tag{9}$$

where the distribution of random variables is given by

$$p(y_t = a \mid \nu < t) = f_0(z), \qquad z \in \Sigma, \tag{10}$$
$$p(y_t = a \mid \nu \geq t) = f_1(z), \qquad z \in \Sigma, \tag{11}$$
$$p(\eta = t | \eta \geq t) = \phi(y_1 \ldots y_t), \tag{12}$$
$$p(\nu = t) = \rho(1 - \rho)^{t-1}. \tag{13}$$

The problem of quickest change detection can be modeled as an indefinite-horizon POMDP with

$$\begin{aligned} \mathcal{S} &= \{\text{PreChange}, \text{PostChange}, \text{PostAlarm}\}, \\ \mathcal{A} &= \{\text{Alarm}, \text{NoAlarm}\}, \\ c(\text{PreChange}, \text{Alarm}) &= 1, \\ r(\text{PostChange}, \text{NoAlarm}) &= 1, \end{aligned} \tag{14}$$

and constraint functional

$$E_\Phi \left[ \sum_{t=1}^{T} c(a_t, s_t) \right] \leq \alpha. \tag{15}$$

State transition and observation probabilities are as shown in Figure 1.

### Structural Monitoring and Maintenance

Bridges, aircraft airframes, and other structural components in engineered systems are subject to fatigue failures. Nondestructive techniques for inspection of the damage state of the component are imperfect, so the true structural damage state of the component is imperfectly knowable. The maintainer may take several actions in structural monitoring, including inspection, repair, or replacement. Because the consequences of structural failure are often severe, the structural monitoring and maintenance problem is most naturally formulated as an infinite-horizon discounted POMDP in which the objective is to minimize lifecycle maintenance and replacement costs subject to a constraint on the probability of failure, expressed at least qualitatively in terms like

"no more than one in a million". Because the service life of the engineered system is usually uncertain, a discounted life cycle cost objectives and occupation measure constraints are appropriate (Puterman 2005, p. 125).

### Opportunistic Spectrum Access

Limits on bandwidth for wireless services motivate dynamic spectrum sharing strategies such as opportunistic spectrum access (OSA), which is a strategy to allow secondary users to exploit instantaneous spectrum opportunities while limiting the level of interference for primary users. Zhao and Swami proposed a CPOMDP decision-theoretic framework for solution of the OSA problem (Zhao & Swami 2007). In this framework, the state is determined by whether or not the spectrum is in use by the primary user, the observations consist of partial spectrum measurements, the reward function is the number of bits delivered by the secondary user, and the constraint is on the probability of collision with a primary user.

## Exact Dynamic Programming Update for CPOMDP

Having defined a CPOMDP and provided some motivating problems, we now describe how to extend dynamic programming techniques for POMDPs to solve CPOMDPs.

A POMDP can be transformed into an infinite-state fully observed Markov decision process by defining a new *belief state*, which is a vector $b \in [0,1]^{|S|}, \sum b(s) = 1$, representing the posterior probability that the system is in each state $s \in S$, given the starting state, observation history, and action history. The belief state is a *sufficient statistic* and can be recursively updated via Bayes rule. If observation $z$ is made and action $a$ taken, and if the previous belief state vector was $b(s)$, the new belief state vector $b'(s')$ is given by

$$b'(s') = \frac{p(z|s') \sum_s p(s'|a,s) b(s)}{\sum_{s,s''} p(z|s'') p(s''|a,s) b(s)}. \quad (16)$$

The dynamic programming update for a POMDP is

$$v^{n+1}(b) = \min_{a \in A} \left[ r(b,a) + \lambda \sum_{z \in \Sigma} p(z|b,a) v^n(b'(b,a,z)) \right].$$

Exact methods for efficiently performing the dynamic programming update for POMDPs include incremental pruning (Cassandra 1998) and restricted region incremental pruning (Feng & Zilberstein 2004). A key technique for both of these methods is representation of the value function as a piecewise-linear function over the belief simplex. This representation allows for a finite representation of the value function as a set of *value function vectors,* the elements of which represent the intersection of a hyperplane with a given axis of the belief simplex. Stationary nonrandomized policies resulting from a stationary value function are sufficient for constrained discounted (Piunovskiy & Mao 2000) and indefinite-horizon (Feinberg & Piunovskiy 2000) Markov decision processes problems with continuous state space.

Exact techniques for the POMDP dynamic programming update implicitly enumerate all value function vectors for the updated value function, and use a pruning operation to maintain a minimal representation. An explicitly enumerative dynamic programming update to convert a set of value function vectors $\mathcal{V}$ to a new set $\mathcal{V}'$ is

$$\mathcal{V}' = \cup_{a \in A} \oplus_{z \in \Sigma} \{v^{a,z,i} | v^i \in \mathcal{V}\} \quad (17)$$

with

$$v^{a,z,i}(s) = \frac{r(s,a)}{|\Sigma|} + \lambda \sum_{s' \in S} p(z|s,a) p(s',s,a) v^i(s'), \quad (18)$$

and $\lambda = 1$ for the indefinite-horizon problem or $0 < \lambda < 1$ for the infinite-horizon discounted problem. Typically, many of the value function vectors in $\mathcal{V}'$ are dominated and superfluous to a minimal value function representation. Pruning is the process of identifying vectors that are not part of the minimal value function representation. At the heart of the pruning operation is solution of a linear program to determine if there is a point where a given value function vector $w$ is better (lower cost) than all other value function vectors $u^i \in \mathcal{U}$. The linear program takes the form

variables: $h \geq 0, b(s) \geq 0 \; \forall s \in S$
maximize: $h$
subject to the constraints:
$b \cdot (u - w) \geq h, \forall u \in \mathcal{U}$
$\sum_{s \in S} b(s) = 1.$

If the linear program is feasible, then $w$ is a candidate for the minimal value function representation. Incremental pruning interleaves pruning with vector generation, to reduce the computational cost associated with pruning for each dynamic programming update. With $\mathbb{PR}$ representing the pruning operation, the dynamic programming update is given by

$$\mathcal{V}' = \mathbb{PR}(\cup_{a \in A} \mathcal{V}^a),$$
$$\mathcal{V}^a = \mathbb{PR}(\mathcal{V}^{a,z_1} \oplus \mathbb{PR}(\mathcal{V}^{a,z_2} ... \mathbb{PR}(\mathcal{V}^{a,z_{k-1}} \oplus \mathcal{V}^{a,z_k}) ...)),$$
$$\mathcal{V}^{a,z} = \mathbb{PR}(\{v^{a,z,i} | v^i \in \mathcal{V}\}). \quad (19)$$

This dynamic programming technique can be extended to the constrained problem by generating two sets of value function vectors with a one-to-one correspondence among the vectors in the set. One set is valued with respect to the cost function $r$, and the other is valued with respect to the constraint function $c$. We write $(v_r^i, v_c^i)$ for the vectors corresponding to each other in the two sets, and $\mathcal{V}$ for the collection of pairs. Let

$$v_r^{a,z,j}(s) = \frac{r(s,a)}{|\Sigma|} + \lambda \sum_{s' \in S} p(z|s,a) p(s',s,a) v_r^j(s') \quad (20)$$

and

$$v_c^{a,z,j}(s) = \frac{c(s,a)}{|\Sigma|} + \lambda \sum_{s' \in S} p(z|s,a) p(s',s,a) v_c^j(s'). \quad (21)$$

Equation 17 is applied separately to $\{v_r^{a,z,j}(s)\}$ and $\{v_c^{a,z,j}(s)\}$, and vectors $(v_r^i, v_c^i)$ in the resulting sets correspond if they were generated from the same action and the same predecessor vector index for each observation.

To develop a pruning operation for the value function vectors, note that for a vector $v$ to belong to the minimal set, it

must have a lower cost $v_r \cdot b$ at $b$ than any another vector $u_r^i \in \mathcal{U}$ that satisfies the constraint $u_c^i \cdot b < \alpha$ for a pair $(u_r, u_c)$. This concept is illustrated in Figure 2. If a vector $u^i$ that does not satisfy the constraint $u_c^i \cdot b < \alpha$ at $b$, the test vector $w$ need not have a lower cost. Thus there are conditional constraints. A standard trick for incorporating conditional constraints into linear programs is to add a discrete variable $d^i \in \{0, 1\}$, resulting in a mixed integer linear program. Applying this technique to the problem at hand produces the mixed integer linear program

variables:
$h \geq 0, b(s) > 0 \; \forall s \in S$,
$d^i \in \{0, 1\} \; \forall i \in \{1, \ldots, |\mathcal{U}|\}$
maximize: $h$
subject to the constraints:
$b \cdot v_c \leq \alpha$,
$b \cdot (u_r^i - v_r) - h \geq -d^i M$,
$u_c^i \cdot b \geq d^i \alpha, \; \forall (u_c^i, u_r^i) \in \mathcal{U}$,
$\sum_{s \in S} b(s) = 1$,

with $M$ a large positive number. If $d^i = 1$, then $u_s^i$ violates the constraint at $b$, and a candidate vector $v_r$ need not have a lower cost than $u_r^i$ at $b$. On the other hand, if $d^i = 0$, then $u_s^i$ satisfies the constraint at $b$, and a candidate vector $v_r$ must have a lower cost than $u_r^i$ at $b$. If the program is feasible, then $v_r$ is a candidate for the lowest cost, minimal value function representation satisfying the constraint. The entire pruning operation for a CPOMDP is presented in Table 1.

The CPOMDP pruning operation can be used to perform value iteration using incremental pruning dynamic programming updates using Equation 19. After a dynamic programming update producing a minimal set of vectors $\mathcal{V}$, the value function is given by $v(b) = \min(b \cdot v_r : b \cdot v_c \leq \alpha, (v_r, v_c) \in \mathcal{V})$. The value function converges to the optimal value function with each successive dynamic programming update.

As an alternative to value iteration, the exact dynamic programming update can be used to perform policy iteration usings Hansen's algorithm (Hansen 1998), which uses a deterministic finite-state controller $(Q, q_o, \Sigma, \delta, \mathcal{A}, \alpha)$ for a policy representation, where

- $Q$ is a finite set of *controller states,*one for each vector $v \in \mathcal{V}$,
- $q_0 \in Q$ is the *start state,*
- $\Sigma$ is a finite *input alphabet,*
- $\delta$ is a function from $Q \times \Sigma$ into $Q$, called the *transition function,*
- $\mathcal{A}$ is a finite *output alphabet,*
- $\alpha$ is an *output function* from $Q$ into $\mathcal{A}$, with $\alpha(i)$ as shorthand for $\alpha(q_i)$.

Because the combination of the system with the controller yields a Markov chain, any policy can be evaluated via solution of a linear system with coefficient matrix $(I - \lambda A)$ with $I$ an identity matrix, $A$ the system/controller transition matrix, and $\lambda = 1$ for the indefinite-horizon problem or $0 < \lambda < 1$ for the infinite-horizon problem. The system state index $i$ and the controller state index $j$ are mapped to a new index $k$ via a one-to-one mapping $(i, j) \leftrightarrow k$, and the

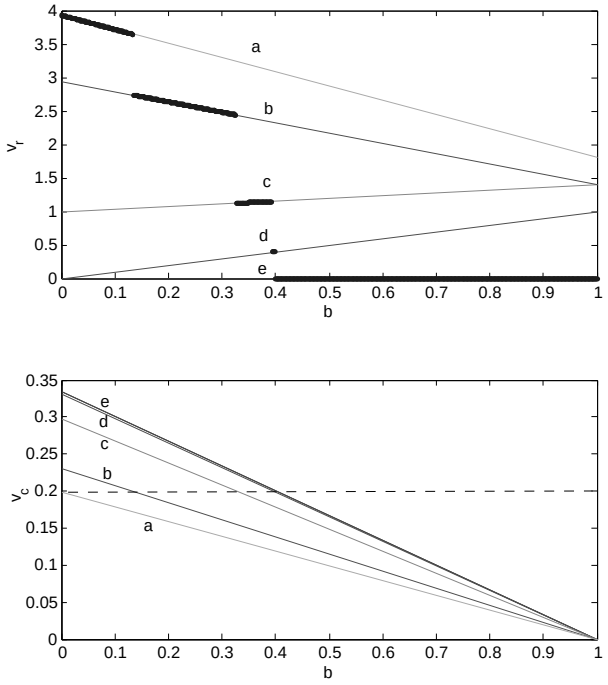


Figure 2: The top plot shows value functions vectors with objective function valuation $v_r$, while the bottom plot shows value function vectors with constraint valuation $v_c$. Vectors in the top plot are highlighted in the region of the belief space where the vector is the best (lowest cost) vector satisfying the constraint $\alpha = 0.20$.

coefficients of the controller/system transition matrix $A$ are given by

$$A\left((i,j),(i',j')\right) \Leftarrow \sum_{z \in \Sigma} p(i'|i, \alpha(j)) p(z|j, \alpha(j)) \mathbf{1}\{\delta(j, z) = j'\}. \quad (22)$$

The linear system for finding the objective value function of a policy is $(I - \lambda A)r = x$, where $r_{(i,j)} = r(i, \alpha(j))$ and $x_{(i,j)}$ is the value function intercept for the $i$th belief state vertex for the $j$th value function vector $v_r^j$. The linear system for finding the constraint value function of a policy is $(I - \lambda A)c = x$, where $c_{(i,j)} = c(i, \alpha(j))$ and $x_{(i,j)}$ is the value function intercept for the $i$th belief state vertex for the $j$th value function vector $v_c^j$.

Under Hansen's policy iteration algorithm, an initial policy is evaluated with respect to both the objective and constraint functions, and then a dynamic programming update is performed using Equation 19 and the pruning operation described in Table 1. Each vector in the updated value function represents a new candidate state for the finite state controller. Once new states have been added, the policy is evaluated to produce a new value function, and the algorithm repeats. Each iteration yields an improved finite state controller that satisfies the problem constraints.

Table 1: Algorithm for pruning value function vectors for a CPOMDP

**Procedure:** MIP-DOMINATE($(w_r, w_c), \mathcal{U}$)
solve the following mixed integer linear program:
variables:
  $h \geq 0, b(s) > 0 \; \forall s \in S$,
  $d^i \in \{0, 1\} \; \forall i \in \{1, \ldots, |\mathcal{U}|\}$
maximize: $h$
subject to the constraints:
  $b \cdot w_c \leq \alpha$,
  $b \cdot (u_r^i - w_r) - h \geq -d^i M$,
  $u_c^i \cdot b \geq d^i \alpha, \; \forall (u_r^i, u_c^i) \in \mathcal{U}$
  $\sum_{s \in S} b(s) = 1$.
**if** $h \geq 0$ **then**
  **return** $b$
**else**
  **return** $nil$
**end if**

**Procedure:** BEST($b$,$\mathcal{U}$)
$min \leftarrow \infty$
**for all** $(u_r, u_c) \in \mathcal{U}$ **do**
  **if** $b \cdot u_c < \alpha$ **then**
    **if** $(b \cdot u_r < min)$ or $((b \cdot u_r = min)$ and $(u_r <_{lex} w_r))$ **then**
      $(w_r, w_c) \leftarrow (u_r, u_c)$
      $min \leftarrow b \cdot u$
    **end if**
  **end if**
**end for**
**return** $(w_r, w_c)$

**Procedure:** $\mathbb{PR}(\mathcal{W})$
$\mathcal{D} \leftarrow \emptyset$
**while** $\mathcal{W} \neq \emptyset$ **do**
  $(w_r, w_c) \leftarrow$ any element in $\mathcal{W}$
  $b \leftarrow$ MIP-DOMINATE($(w_r, w_c), \mathcal{D}$)
  **if** $b = nil$ **then**
    $\mathcal{W} \leftarrow \mathcal{W} - \{(w_r, w_c)\}$
  **else**
    $(w_r, w_c) \leftarrow$ BEST($b, \mathcal{W}$)
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(w_r, w_c)\}$
    $\mathcal{W} \leftarrow \mathcal{W} - \{(w_r, w_c)\}$
  **end if**
**end while**
**return** $\mathcal{D}$

## Example

The solution technique is illustrated with an example.

Consider the indefinite-horizon change detection problem with geometric change time parameter $\rho = 0.01$, observation alphabet size $|\Sigma| = 3$, pre-change observation distribution

$$f_0(z_1) = 0.6,$$
$$f_0(z_2) = 0.3,$$
$$f_0(z_3) = 0.1,$$

and post-change observation distribution

$$f_0(z_1) = 0.2,$$
$$f_0(z_2) = 0.4,$$
$$f_0(z_3) = 0.4.$$

The problem statement is

$$\underset{\phi \in \Phi}{\text{minimize}} \; v = E[(\tau - \eta - 1)^+] \tag{23}$$

subject to

$$p(\tau < \eta) \leq \alpha. \tag{24}$$

with false alarm probability constraint $\alpha = 0.20$.

Starting with initial value function vectors $v_r = [10 \; 0 \; 0]$ and $v_c = [0 \; 0 \; 0]$, we perform four dynamic programming updates. The results are shown in Figure 3. The policy corresponding to the value function consists of producing an alarm when the probability that the system is in the post-change state exceeds 0.40, with an expected detection delay of approximately 4. Tartakovsky and Veeravalli noted that a conservative policy for the problem of quickest detection consists of producing an alarm when the posterior probability of change exceeds $1 - \alpha$ (Tartakovsky & Veeravalli 2004). Note that this policy is much less conservative (and therefore better-valued) than the conservative policy of producing an alarm when the posterior probability of change exceeds $1 - \alpha = 0.80$.

## Conclusion

This paper presents an exact dynamic update for CPOMDPs using a piecewise linear representation of the value function. The key concept is modification of the standard pruning algorithm, with a mixed integer linear program replacing a linear program. The dynamic programming update can be used to perform value iteration or policy iteration. The method should be useful for finding solutions to modestly-sized sequential decision problems that include uncertainty and constraints.

The scaling behavior of the proposed CPOMDP dynamic programming update is expected to be similar to that of exact dynamic programming updates for POMDPs, which exhibit PSPACE complexity for many problems. Although this scaling behavior makes the proposed technique unsuitable for large-scale problems, we hope that the insight provided by an exact numerical technique will spur development of approximate methods for solution of CPOMDPs.

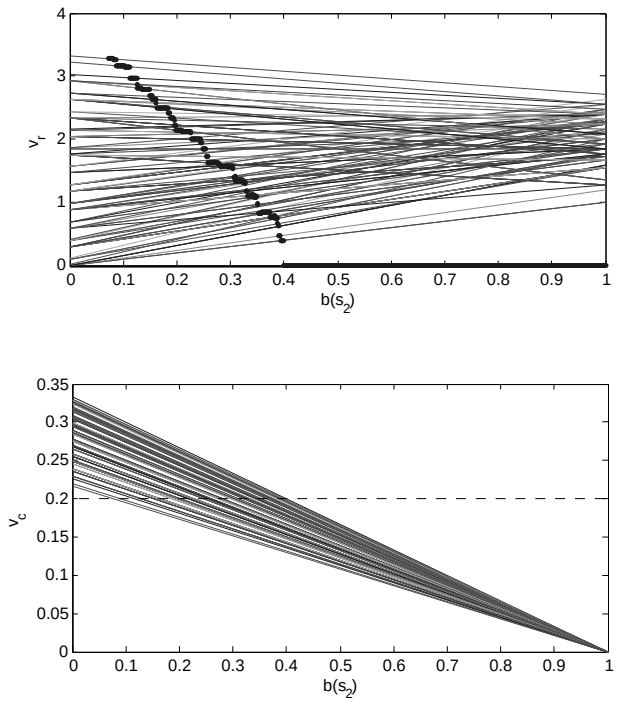


Figure 3: A minimal representation of the value function for the change detection example after four exact dynamic programming updates. Each vector is presented with its objective valuation $v_r$ and constraint valuation $v_c$, with $b(s_2)$ the posterior probability that the system is in the post-change state. An approximation of the optimal value function $v(b)$ is shown as dots on the top plot.

## References


Altman, E. 1999. *Constrained Markov Decision Processes*. Boca Raton, LA: CRC Press.

Cassandra, A. R. 1998. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph.D. Dissertation, Brown University.

Feinberg, E. A., and Piunovskiy, A. B. 2000. Multiple objective nonatomic markov decision processes with total reward criteria. *J. Math. Anal. Appl.* 247:45–66.

Feng, Z., and Zilberstein, S. 2004. Region-based incremental pruning for POMDPs. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 146–153.

Hansen, E. A. 1998. *Finite Memory Control of Partially Observable Systems*. PhD thesis, University of Massachusetts Amherst.

Hansen, E. A. 2007. Indefinite-horizon POMDPs with action-based termination. In *22nd National Conference on Artificial Intelligence*.

Meyn, S. P. 2007. *Control Techniques for Complex Networks*. Cambridge, UK: Cambridge University Press.

Patek, S. D. 2001. On partially observed stochastic shortest path problems. In *40th IEEE Conference on Decision and Control*.

Piunovskiy, A. B., and Mao, X. 2000. Constrained markovian decision processes: the dynamic programming approach. *Operations research letters* 27:119–126.

Poupart, P. 2005. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, University of Toronto.

Puterman, M. L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons.

Shiryaev, A. N. 1963. On optimum methods in quickest detection problems. *Theory of Probability and Its Applications* 8(1):22–46.

Spaan, M. T. J., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research* 24:195–220.

Tartakovsky, A. G., and Veeravalli, V. V. 2004. General asymptotic Bayesian theory of quickest change detection. *Theory of Probability and Its Applications* 49(3):458–497.

Zhao, Q., and Swami, A. 2007. A decision-theoretic framework for opportunistic spectrum access. *IEEE Wireless Communications* 14(4):14–20.