



POMDP'S : Exact and Approximate Solutions

Bharaneedharan R

University of Illinois at Chicago

Road-map

- ⑥ Definition of POMDP
- ⑥ Belief as a sufficient statistic
- ⑥ General solutions to a POMDP
- ⑥ Monahan's enumeration algorithm
- ⑥ Incremental pruning
- ⑥ Approximations
- ⑥ MDP based approximations
- ⑥ Grid based approximation

Definition of POMDP

A POMDP is defined by the tuple $\langle S, A, T, \theta, O, R \rangle$

- ⑥ S is a finite set of world states.
- ⑥ A is a finite set of actions that can be performed by the agent.
- ⑥ $T : S \times A \times S \rightarrow [0,1]$, the likelihood of an action changing the system state from one to another.
- ⑥ θ is the finite set of observations that the agent can make (sensory inputs available for the agent).
- ⑥ $O : S \times \theta \times A \rightarrow [0,1]$, the likelihood of making a certain observation at a state after performing a particular action.
- ⑥ $R : S \times A \times S \rightarrow \mathfrak{R}$, defines payoff's for the agent in a given system state after performing an action.

Belief's and Information States

- ⑥ Let the information state process(ISP) be I_0, I_1, \dots, I_t
- ⑥ $I_t = [o_t, a_{t-1}, I_{t-1}]$
- ⑥ Belief or the probability distribution over states at any time t is given as $Be_j(t) = P(S(t) = j \mid I_{t-1})$
- ⑥ I_0 is the information state before agents start performing actions.
- ⑥ If observations result only after performing actions $I_0 = Be(0)$, the prior at time $t=0$
- ⑥ Redefining ISP $I_0 = Be(0), I_1 = [o_1, a_0, I_0], \dots$

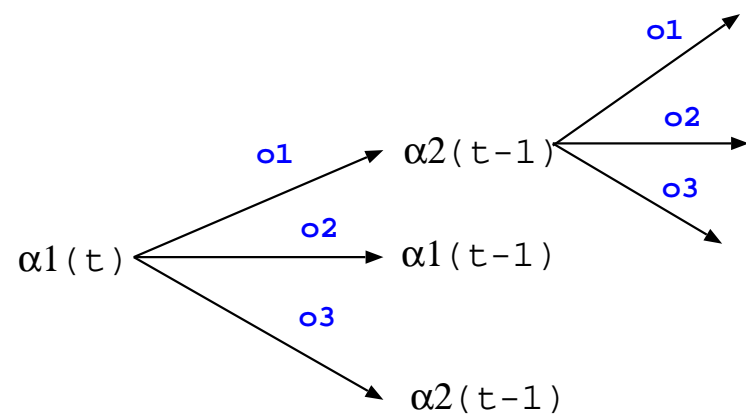
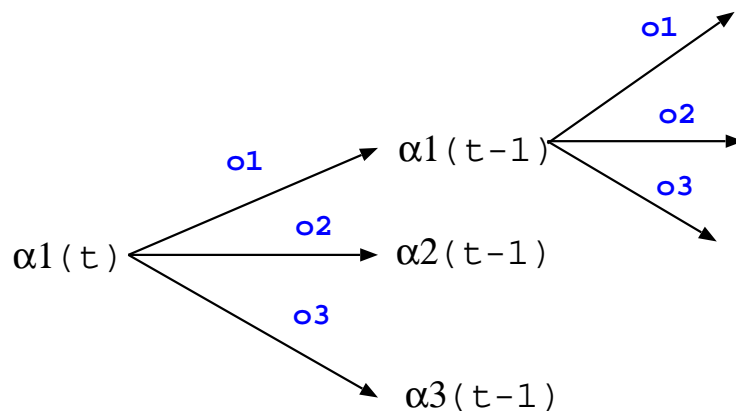
Belief is a sufficient statistic

- ⑥ $Be_j(t) = P(S(t) = j \mid o_t, a_{t-1}, I_{t-1})$
- ⑥ At $t=0$, $Be(0)$ is just a prior
- ⑥ At $t=1$, $Be_j(1) = P(S(t) = j \mid o_1, a_0, Be_j(0))$
- ⑥ We see that at any time t , $Be(t)$ will be dependent on current observation, previous action and previous belief state.
- ⑥ Also current observation and state is dependent on only previous action and information state
- ⑥ Consequently $Be(t)$ is a compact representation of I_t

General solution to a POMDP

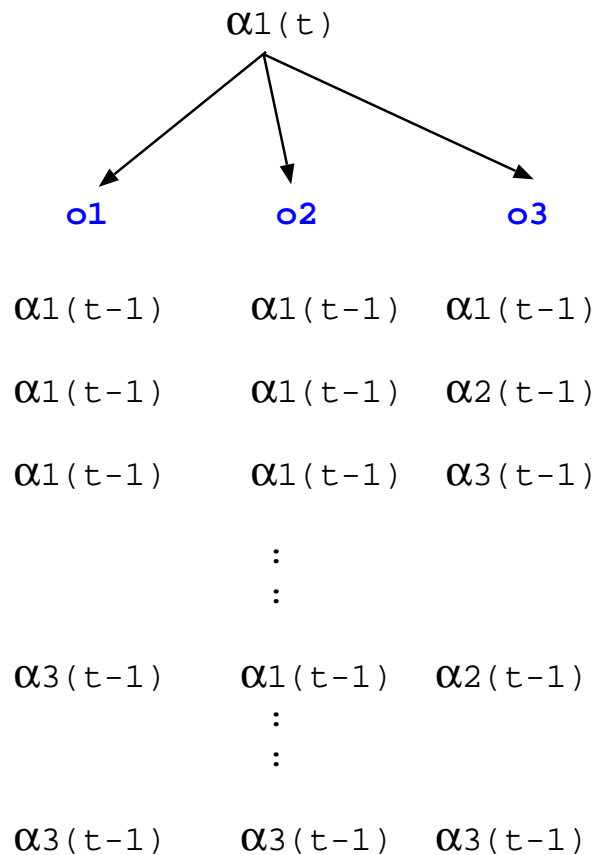
Let $b_i \in \Delta(S)$ for all i

- ⑥ $V_t(b_i) = \text{Max}_{a \in A} R(b_i | a) + \gamma \sum_{o \in \theta} P(o | b_i, a) \times V_{t-1}(\tau(b_i, o, a))$
- ⑥ $\alpha_t(s_i, a) = R(s_i, a) + \gamma \sum_{o \in \theta} \sum_{s_j \in S} P(o | s_j, a) \times P(s_j | s_i, a) \times \alpha'_{t-1}$
- ⑥ α'_{t-1} is an alpha vector from the previous epoch for a given observation o .



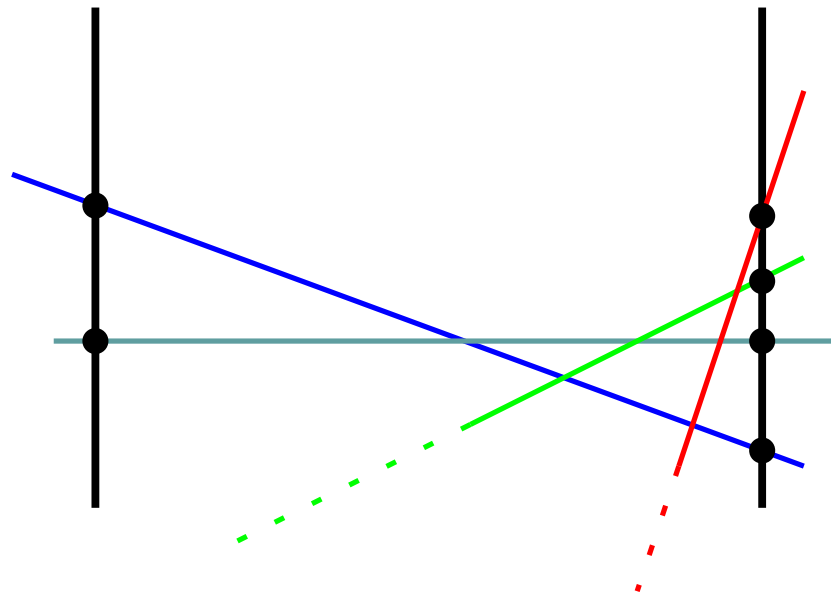
Continued ...

- ⑥ Each choice of α'_{t-1} for a given o is a choice of a future plan given o .
- ⑥ α_t is constructed using the best of such choices of plans for all possible observations.



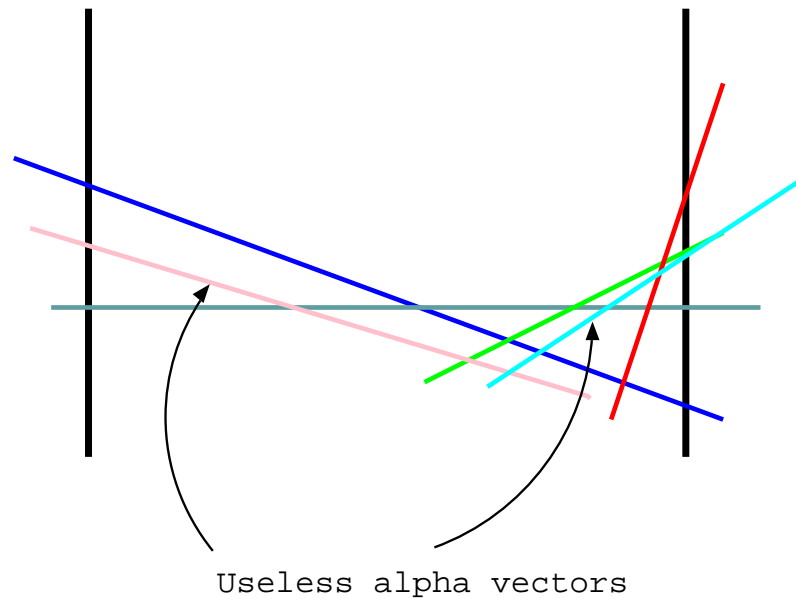
Piecewise linearity and consequences

- ⑥ We know that the value function is piecewise linear i.e. $V_t(b_i) = \text{Max}_{\alpha_t^k} \alpha_t^k \cdot b_i$
- ⑥ Computing the value function involves just projecting the alpha vectors.
- ⑥ But alpha vectors are linear over the belief space and hence only the values simplex corners represented in alpha vectors need to be computed.



Enumeration algorithm [Monahan 82]

- ⑥ At a given epoch, project all the alpha vectors describing the value function at previous epoch.
- ⑥ If we have Γ_t is the set of alpha vectors at epoch t, $|\Gamma_t| = |A| |\Gamma_{t-1}|^{|\Theta|}$
- ⑥ Find the useful set of alpha vectors that forms the value function



Testing for redundancy

- ⑥ Test each alpha vector if it maximizes the value function at least at one point in the belief space
- ⑥ Let $\pi_i \in \pi = Be(s = i)$ and α_k be the vector to be tested
- ⑥ α_k is an useful vector iff
for all vectors $j \neq k$, $\sum_i \pi_i \alpha_j \leq \sum_i \pi_i \alpha_k$ is satisfied at some π
- ⑥ i.e. $\sum_i \pi_i (\alpha_j - \alpha_k) \leq 0$
- ⑥ To find the point where the vector maximizes the most we rewrite the above as an LP
maximize : δ
 $\sum_i \pi_i (\alpha_j - \alpha_k) + \delta \leq 0$ for each alpha vector $j \neq k$
 $\sum_i \pi_i = 1$
 $\pi_i \geq 0$
- ⑥ This leads to an LP with $(|\Gamma| - 1) + 1 + |S|$ constraints and $|S| + 1$ variables

Problems with simple enumeration

- ⑥ Brute force generate and test
- ⑥ For a problem with 30 states, 4 actions and 8 observations, we would need about 60MB of memory to store the new vectors in epoch 2.
- ⑥ With 12 observations that's a whopping 15GB.
- ⑥ Useful only for solving very small problems

Incremental pruning [zhang, Liu 96]

- ⑥ Extension to the enumeration algorithm, using interleaved generation and redundancy testing.
- ⑥ The key is in breaking up the dynamic programming update of the value function.

$$⑥ \quad V(b) = \max_{a \in A} V^a(b)$$

$$V^a(b) = \sum_{o \in \theta} V_o^a(b)$$

$$V_o^a(b) = \frac{\sum_s R(a,s)b(s)}{|\theta|} + \gamma P(o | b, a) V(b_o^a)$$

- ⑥ The maximizing set of alpha vectors is determined from bottom to top.
- ⑥ Leading to iterative purging of smaller set of alpha vectors

Partial projection and pruning

Let W', W^a, W_o^a be the set of vectors describing V', V^a, V_o^a

$$\textcircled{6} \quad W' = \text{purge} \left(\bigcup_{a \in A} W^a \right)$$

$$\textcircled{6} \quad W^a = \text{purge} \left(\bigoplus_{o \in \theta} W_o^a \right)$$

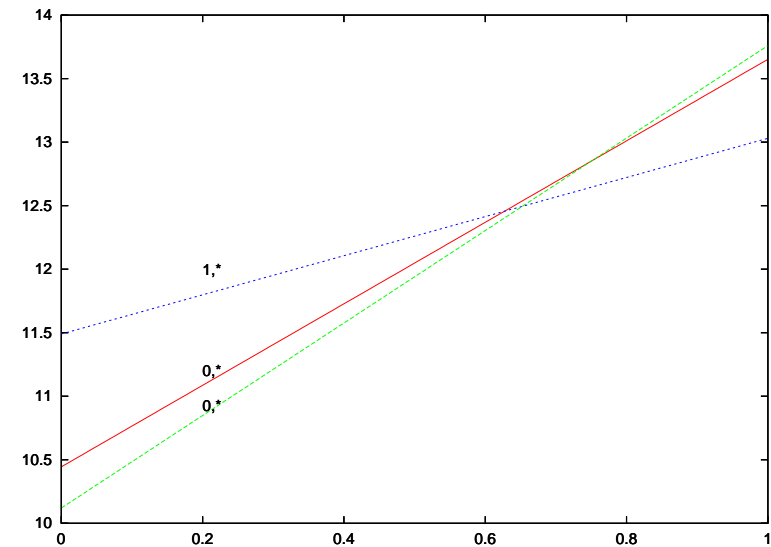
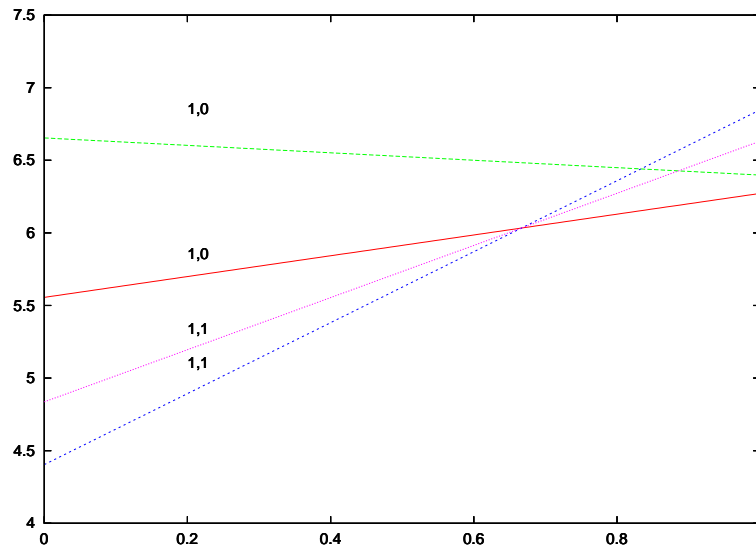
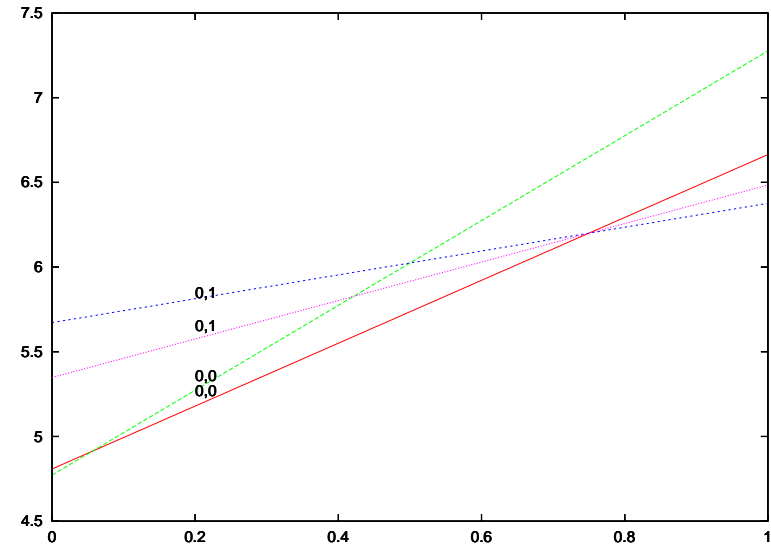
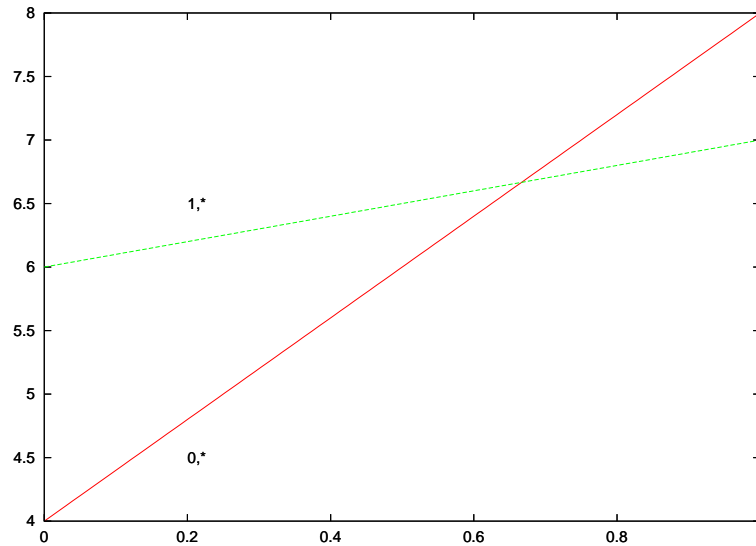
$$\textcircled{6} \quad W_o^a = \text{purge} \left(\{ \tau(\alpha, a, o) \mid \alpha \in W \} \right)$$

$$\textcircled{6} \quad W' = \text{projection}(W)$$

$$\textcircled{6} \quad \tau(\alpha, a, o)(s) = \frac{R(a, s)}{|\theta|} + \gamma \sum_{s'} \alpha(s') P(o \mid s', a) P(s' \mid s, a)$$

$\textcircled{6}$ $\text{purge}(\cdot)$ returns the maximizing set of vectors in the belief space ($|S|$ dimensional space)

Example

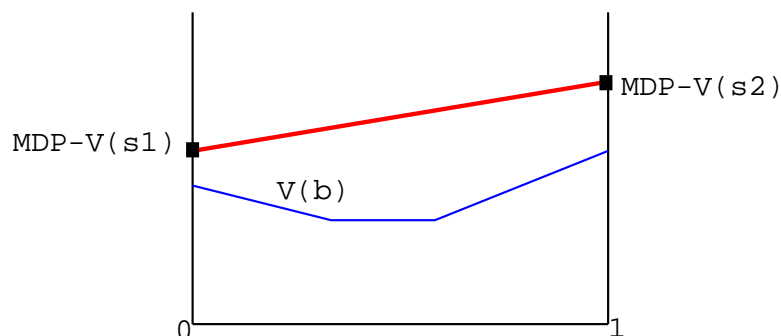


Approximations

- ⑥ MDP based approximation
- ⑥ Grid based approximation

MDP based approximation

- ⑥ Is a crude approximation assuming complete observability
- ⑥ Solve the underlying MDP and compute value function V_{MDP}^*
- ⑥ Value of a belief state is computed as $V(b) = \sum_s b(s)V_{MDP}^*(s)$
- ⑥ We can also redefine our update rule as
$$V_{i+1}(b) = \sum_{s'} b(s') \max_{a \in A} [R(s, a) + \gamma \sum_s P(s | s', a) V_i^{MDP}(s)]$$
- ⑥ Note V_i will always be described by a single vector
- ⑥ The MDP based update above provides an upper bound to the general update using observations.



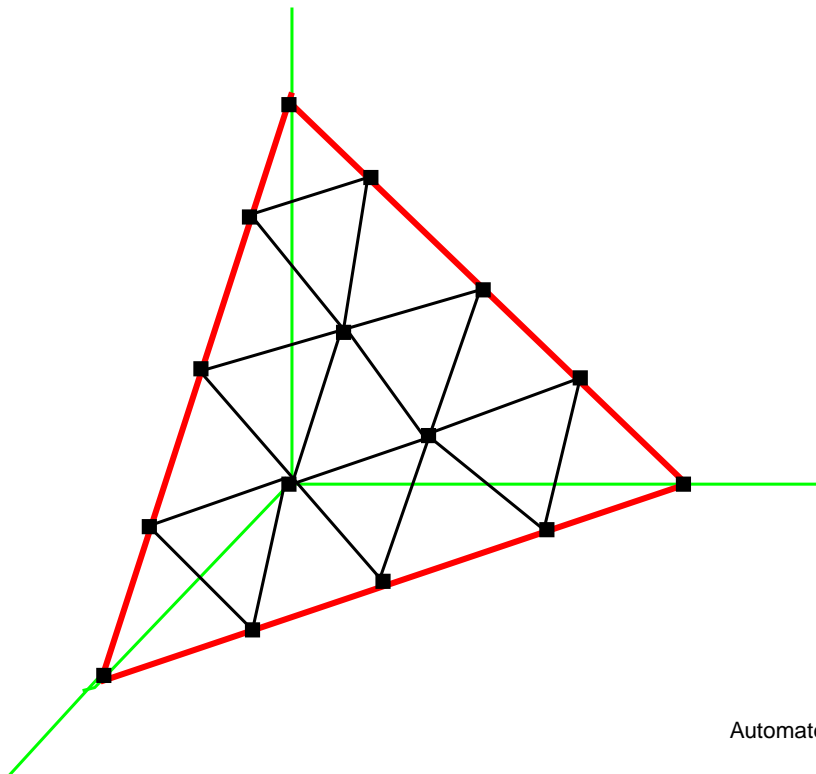
Upper bound property

- ⑥ Let H be our regular POMDP update function involving observations.
- ⑥ H_{MDP} is the MDP based update function.
- ⑥ We have to show that $HV_i \leq H_{MDP} V_i$

$$\begin{aligned} HV_i(b) &= \max_{a \in A} \sum_{s \in S} R(s, a) b(s) + \gamma \sum_{o \in \theta} \sum_{s' \in S} \sum_{s \in S} P(s', o | s, a) b(s) \alpha_i^{MDP}(s') \\ &= \max_{a \in A} \sum_{s \in S} b(s) [R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_i^{MDP}(s')] \\ &\leq \sum_{s \in S} b(s) \max_{a \in A} [R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_i^{MDP}(s')] = H_{MDP} V_i(b) \end{aligned}$$

Grid based approximation [Lovejoy 91]

- ⑥ General value iteration in all exact algorithms compute the value function over the complete belief space.
- ⑥ Compute the value function at finite number of points in the belief space.
- ⑥ Compute the value of other belief states with values of our chosen set of belief states using interpolation.



Grid based value function updates

- ⑥ Let G be the set of grid points
- ⑥ $V(b) = E(b, V_G) = \sum_{j=1}^{|G|} \lambda_j V_G(b_j)$
- ⑥ $0 \leq \lambda_j \leq 1$
- ⑥ $\sum_{j=1}^{|G|} \lambda_j = 1$
- ⑥ $V_{i+1}(b_j^G) = \max_{a \in A} R(b_j^G, a) + \gamma \sum_{o \in \theta} P(o | \tau(b_j^G, a), a) V_i(\tau(b_j^G, a, o))$
- ⑥ Let us call this update function based on grid points as H_G

Lower bounds

- ⑥ The grid based update function H_G provides a lower bound to the regular update function H .
- ⑥ $H_G V_i \leq H V_i$
- ⑥ Proof: Let V_i be a set of vectors describing the value function

H_G will always use only a partial set of vectors from V_i for updating the value of the grid points.

Since H_G only considers maximal vectors at grid points it ignores the maximal vectors at other belief points.

In the best case scenario $H_G V_i = H V_i$

Upper bounds

- ⑥ The value function based on the grid points and point interpolation together provides us with an upper bound on the value function.

$$HV(b) = HV \left(\sum_{i=1}^{|G|} \lambda_i b_i^G \right)$$

$$\leq \sum_{i=1}^{|G|} [HV(b_i^G)] = H_G V(b)$$