Planning, Execution & Learning: Planning with POMDPs

Reid Simmons

Markov Models

- A Markov node represents complete state of the world
- ***Markov Property*** implies that current state is sufficient

- Plan is a ***Policy***
  - ***Stationary*** policy: Best action is fixed
  - ***Non-stationary*** policy: Best action depends on time

- Categories of Markov Models

	Passive	**Choose Actions**
Fully Observable	**Markov Models**	**MDP**
Hidden State	**HMM**	**POMDP**

Tradeoffs

- **MDP**
  - + Tractable to solve
  - + Relatively easy to specify
  - – Assumes perfect knowledge of state

- **POMDP**
  - + Treats all sources of uncertainty (action, sensing, environment) in a uniform framework
  - + Allows for taking actions that gain information
  - – Difficult to specify all the conditional probabilities
  - – *Hugely* intractable to solve optimally

POMDP Models

- What is a **POMDP**?
  - Basically an **MDP**, except that state is not known with certainty
    - Probability distribution (belief state) over world states
  - Model sensors using conditional probabilities
    - $p_i(o \mid s, a)$
    - $p_{left}(small\_opening \mid junction, move\_forward) = 0.20$
  - Action update rule:
    - $p_{posterior}(s) = \sum_{s' \in S,\, a \in A(s)} p(s|a, s') \cdot p(s')/k$
  - Observation update rule:
    - $p_{posterior}(s) = p_i(o \mid s, a) \cdot p(s)/k'$

POMDP Conversion

- Equivalent MDP Model
  - Each MDP state is probability distribution (continuous belief state b) over the states of the original POMDP
  - State transitions are product of actions and observations

$$p(s' \mid a, o, b) = p(o \mid s', a, b) \cdot p(s' \mid a, b) / p(o \mid a, b)$$

$$p(o \mid s', a, b) = p(o \mid s')$$

$$p(s' \mid a, b) = \sum_{s \in S} p(s' \mid a, s) \cdot b(s)$$

$$p(o \mid a, b) = \sum_{s' \in S} \left(p(o \mid s') \cdot \sum_{s \in S} p(s' \mid a, s) \cdot b(s) \right)$$

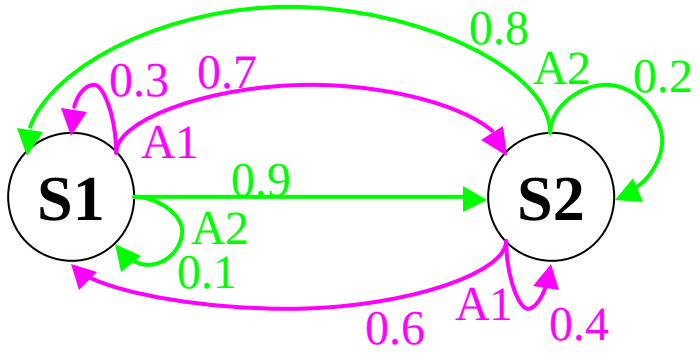
$$= \sum_{s' \in S} p(o \mid s') \cdot p(s' \mid a, b)$$

  - MDP rewards are expected rewards of original POMDP

$$R(a, b) = \sum_{s \in S} r(a, s) \cdot b(s)$$

POMDP Conversion Example (I)

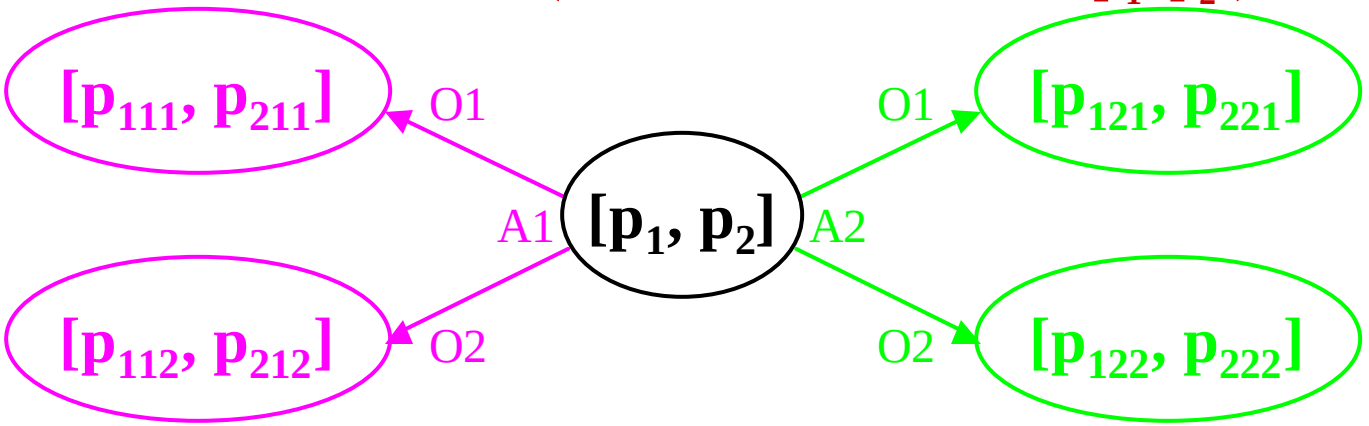
Original POMDP



r(A1, S1) = 2
r(A2, S1) = 1
r(A1, S2) = 1
r(A2, S2) = 3

p(O1 | S1) = 0.9
p(O2 | S1) = 0.1
p(O1 | S2) = 0.5
p(O2 | S2) = 0.5

Transformed MDP (Parametric Form: $b = [p_1, p_2]$)



POMDP Conversion Example (II)

- ***Transformed Rewards***

$R(A1, b) = 2p_1 + p_2 = 2p_1 + (1 - p_1) = (p_1 + 1)$

$R(A2, b) = p_1 + 3p_2 = p_1 + 3(1 - p_1) = (3 - 2p_1)$

- ***Transformed Action Transition Probabilities***

$$p(S1 \mid A1, b) = p(S1 \mid A1, S2) \cdot p(S1) + P(S1 \mid A1, S2) \cdot p(S2)$$
$$= 0.3p_1 + 0.6p_2 = 0.3p_1 + 0.6(1 - p_1)$$
$$= 0.6 - 0.3p_1$$

$p(S2 \mid A1, b) = 0.7p_1 + 0.4p_2 = 0.4 + 0.3p_1$

$p(S1 \mid A2, b) = 0.1p_1 + 0.8p_2 = 0.8 - 0.7p_1$

$p(S2 \mid A2, b) = 0.9p_1 + 0.2p_2 = 0.2 + 0.7p_1$

POMDP Conversion Example (III)

- ***Transformed Observation Probabilities***

$$p(O1 \mid A1, b) = p(O1 \mid S1) \cdot p(S1 \mid A1, b) + P(O1 \mid S2) \cdot p(S2 \mid A1, b)$$
$$= 0.9(0.6 - 0.3p_1) + 0.5(0.4 + 0.3p_1)$$
$$= 0.74 - 0.12p_1$$

$$p(O2 \mid A1, b) = p(O2 \mid S1) \cdot p(S1 \mid A1, b) + P(O2 \mid S2) \cdot p(S2 \mid A1, b)$$
$$= 0.1(0.6 - 0.3p_1) + 0.5(0.4 + 0.3p_1)$$
$$= 0.26 + 0.12p_1$$

$$p(O1 \mid A2, b) = p(O1 \mid S1) \cdot p(S1 \mid A2, b) + P(O1 \mid S2) \cdot p(S2 \mid A2, b)$$
$$= 0.9(0.8 - 0.7p_1) + 0.5(0.2 + 0.7p_1)$$
$$= 0.82 - 0.28p_1$$

$$p(O2 \mid A2, b) = p(O2 \mid S1) \cdot p(S1 \mid A2, b) + P(O2 \mid S2) \cdot p(S2 \mid A2, b)$$
$$= 0.1(0.8 - 0.7p_1) + 0.5(0.2 + 0.7p_1)$$
$$= 0.18 + 0.28p_1$$

POMDP Conversion Example (IV)

- ***State Transition Probabilities (Actions and Observations)***

$p(s \mid a, o, b) = p(o \mid s) \cdot p(s \mid a, b) / p(o \mid a, b)$

$p(S1 \mid A1, O1, b) = p(O1 \mid S1) \cdot p(S1 \mid A1, b) / p(O1 \mid A1, b)$
$= 0.9(0.6 - 0.3p_1)/(0.74 - 0.12p_1)$
$= (0.54 - 0.27p_1)/(0.74 - 0.12p_1)$

$p(S2 \mid A1, O1, b) = (0.20 + 0.15p_1)/(0.74 - 0.12p_1)$
$p(S1 \mid A2, O1, b) = (0.72 - 0.63p_1)/(0.82 - 0.28p_1)$
$p(S2 \mid A2, O1, b) = (0.10 + 0.35p_1)/(0.82 - 0.28p_1)$
$p(S1 \mid A1, O2, b) = (0.06 - 0.03p_1)/(0.26 + 0.12p_1)$
$p(S2 \mid A1, O2, b) = (0.20 + 0.15p_1)/(0.26 + 0.12p_1)$
$p(S1 \mid A2, O2, b) = (0.08 - 0.07p_1)/(0.18 + 0.28p_1)$
$p(S2 \mid A2, O2, b) = (0.10 + 0.35p_1)/(0.18 + 0.28p_1)$

Solving POMDPs

- Maximize Expected Reward Over Belief Space: **V**(***b***)
- Representational Choices
  - Exact **V**, exact ***b***
    - Optimal solutions, but intractable
  - Approximate **V**, exact ***b***
    - Differentiable function approximators (higher-order polynomials, neural nets, …)
  - Exact **V**, Approximate ***b***
    - Dynamic Bayes Net, Particle Filters
  - Approximate **V**, Approximate ***b***
    - Combos of above
- Greedy Approaches Based on Solving Underlying MDP

Exact Solution to POMDP (I)

- Convert to MDP, and Use Value Iteration
  - $V(b) = \max_a \{R(a, b) + \gamma \sum_{b'} p(b' \mid a, b)\, V(b')\}$
- Use Fact that Value Function is ***Piece-Wise Linear Convex***
  - $V(b) = \max_{v \in \Psi} (v \bullet b)$

$p(b' \mid a, b) = p(o \mid a, b)$

$b' = [p(S1 \mid a, o, b), p(S2 \mid a, o, b), \ldots]$

$b' = [p(o \mid S1)p(S1 \mid a, b)/p(o \mid a, b), p(o \mid S2)p(S2 \mid a, b)/p(o \mid a, b), \ldots]$

$\underline{b}' = [p(o \mid S1)p(S1 \mid a, b), p(o \mid S2)p(S2 \mid a, b), \ldots]$

$V(b) = \max_a \{R(a, b) + \gamma \sum_{\underline{b}'} p(o \mid a, b) \max_{v \in \Psi} (v \bullet \underline{b}')/p(o \mid a, b)\}$

$V(b) = \max_a \{R(a, b) + \gamma \sum_{\underline{b}'} \max_{v \in \Psi} (v \bullet \underline{b}')\}$

Exact Solution to POMDP (II)

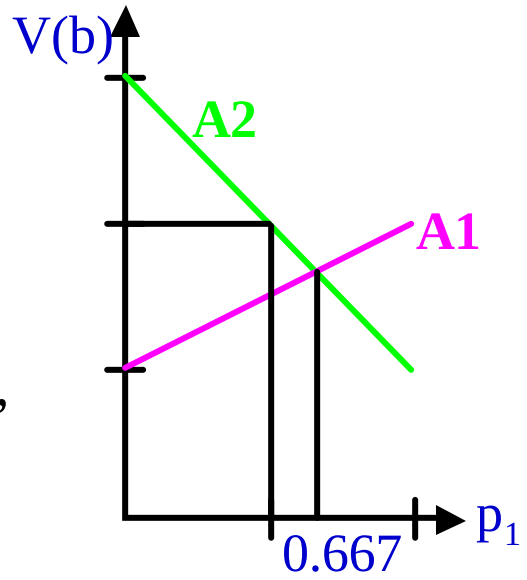
Horizon-Zero Solution: $V_0(b) = 0$

Horizon-One Solution:

$$V_1^{A1}(b) = R(A1, b) + \gamma \cdot 0 = (p_1 + 1)$$
$$V_1^{A2}(b) = R(A2, b) + \gamma \cdot 0 = (3 - 2p_1)$$

$$\Psi_1 = \{[2, 1], [1, 3]\}$$

$$V_1([0.5, 0.5]) = \max([2, 1] \bullet [0.5, 0.5], [1, 3] \bullet [0.5, 0.5])$$
$$= \max(1.5, 2) = 2$$

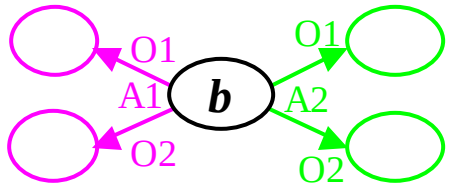


Crossover Point:

$$[2, 1] \bullet [p_1, p_2] = [1, 3] \bullet [p_1, p_2]$$
$$2p_1 + 1(1 - p_1) = p_1 + 3(1 - p_1)$$
$$3p_1 = 2$$

Exact Solution to POMDP (III)

Horizon-Two Solution:



Value Function for Action A1:

$$V_2^{A1}(b) = R(A1, b) + \gamma \cdot \sum_{\underline{b}'} \max_{v \in \Psi} (v \bullet \underline{b}')$$

$$= (p_1 + 1) + \gamma \cdot \{\max_{v \in \Psi} (v \bullet \underline{b}'_1) + \max_{v \in \Psi} (v \bullet \underline{b}'_2)\}$$

$$\underline{b}'_1 = [p(O1 \mid S1)p(S1 \mid A1, b), p(O1 \mid S2)p(S2 \mid A1, b)]$$

$$= [(0.54 - 0.27p_1), (0.20 + 0.15p_1)]$$

$$\underline{b}'_2 = [(0.06 - 0.03p_1), (0.20 + 0.15p_1)]$$

$$V_2^{A1}(b)_a = (p_1 + 1) + \gamma \cdot ([2, 1] \bullet \underline{b}'_1 + [2, 1] \bullet \underline{b}'_2)$$

$$= (p_1 + 1) + \gamma \cdot \{2(0.54 - 0.27p_1) + (0.20 + 0.15p_1) + 2(0.06 - 0.03p_1) + (0.20 + 0.15p_1)\}$$

$$= (p_1 + 1) + \gamma \cdot (1.6 - 0.3p_1)$$

$$V_2^{A1}(b)_b = (p_1 + 1) + \gamma \cdot ([2, 1] \bullet \underline{b}'_1 + [1, 3] \bullet \underline{b}'_2) = (p_1 + 1) + \gamma \cdot (1.94 + 0.03p_1)$$

$$V_2^{A1}(b)_c = (p_1 + 1) + \gamma \cdot ([1, 3] \bullet \underline{b}'_1 + [2, 1] \bullet \underline{b}'_2) = (p_1 + 1) + \gamma \cdot (1.46 + 0.27p_1)$$

$$V_2^{A1}(b)_d = (p_1 + 1) + \gamma \cdot ([1, 3] \bullet \underline{b}'_1 + [1, 3] \bullet \underline{b}'_2) = (p_1 + 1) + \gamma \cdot (1.86 + 0.6p_1)$$

Exact Solution to POMDP (IV)

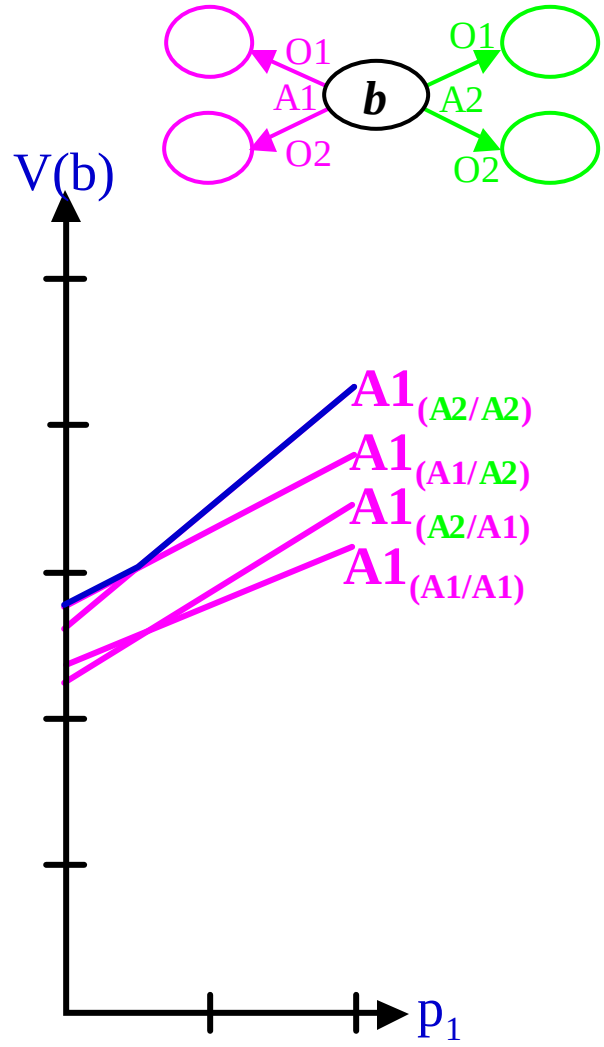
Find Value Vectors (Assume $\gamma = 0.9$):

$V_2^{A1}(b)_a = (p_1 + 1) + \gamma \cdot (1.6 - 0.3p_1)$
$= 2.44 + 0.73p_1$
$= [3.17, 2.44]$

$V_2^{A1}(b)_b = 2.746 + 1.027p_1 = [3.773, 2.746]$

$V_2^{A1}(b)_c = 2.314 + 1.243p_1 = [3.557, 2.314]$

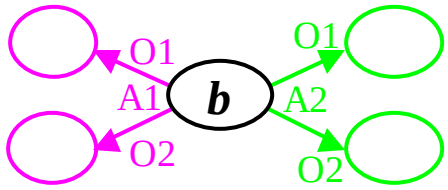
$V_2^{A1}(b)_d = 2.62 + 1.54p_1 = [4.16, 2.62]$

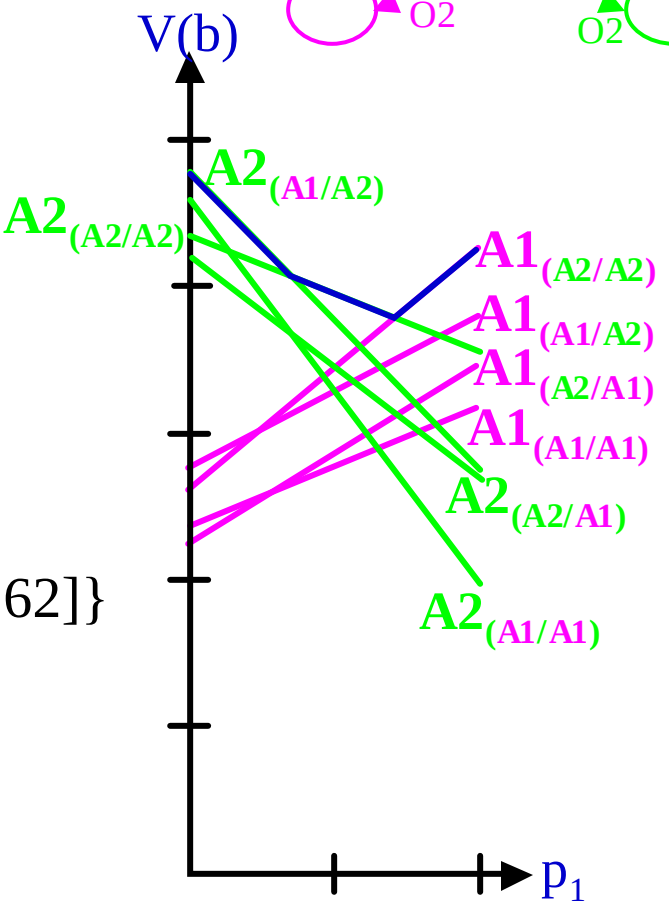


Exact Solution to POMDP (V)

Combining Value Function for Actions A1 and A2

$$V_2^{A2}(b)_a = (3 - 2p_1) + \gamma \cdot (1.8 - 0.7p_1)$$
$$= [1.99, 4.62]$$
$$V_2^{A2}(b)_b = 2.746 + 1.027p_1 = [2.791, 4.728]$$
$$V_2^{A2}(b)_c = 2.314 + 1.243p_1 = [2.719, 4.152]$$
$$V_2^{A2}(b)_d = 2.62 + 1.54p_1 = [3.52, 4.26]$$





Horizon-Two Value Function:

$$\Psi_2 = \{[2.791, 4.728], [3.52, 4.26], [4.16, 2.62]\}$$

Witness Algorithm (Littman, 1994)

- A ***Witness*** is a Counter-Example
  - Idea: Find places where the value function is suboptimal
  - Operates action-by-action and observation-by-observation to build up value vectors
- **Algorithm**
  - Start with value vectors for known ("corner") states
  - Define a linear program (based on Bellman's equation) that finds a point in the belief space where the value of the function is incorrect
  - Add a new vector (a linear combination of the old value function)
  - Iterate

Witness Algorithm: Example

- Choose some belief that has the wrong value
  (by solving system of linear equations)

  Choose: $b_1 = [0.5, 0.5]$

  $V(b_1) = \max([2, 1] \bullet [0.5, 0.5], [1, 3\ ] \bullet [0.5, 0.5]) = \mathbf{2}$

$$V_2^{A1}(b) = (p_1 + 1) + \gamma \cdot \{\max_{v \in \Psi} (v \bullet \underline{b'}_1) + \max_{v \in \Psi} (v \bullet \underline{b'}_2)\}$$
$$= (p_1 + 1) + \gamma \cdot \{\max([2, 1] \bullet [(0.54 - 0.27p_1), (0.20 + 0.15p_1)],$$
$$[1, 3] \bullet [(0.54 - 0.27p_1), (0.20 + 0.15p_1)]) +$$
$$\max([2, 1] \bullet [(0.06 - 0.03p_1), (0.20 + 0.15p_1)],$$
$$[1, 3] \bullet [(0.06 - 0.03p_1), (0.20 + 0.15p_1)])\}$$

$$V_2^{A1}(b) = 1.5 + \gamma \cdot \{\max([2, 1] \bullet [0.405, 0.275], [1, 3] \bullet [0.405, 0.275]) +$$
$$\max([2, 1] \bullet [0.045, 0.275], [1, 3] \bullet [0.045, 0.275])\}$$
$$= 1.5 + \gamma \cdot \{\max(1.085, \underline{\mathbf{1.23}}) + \max(0.365, \underline{\mathbf{0.87}})\}$$

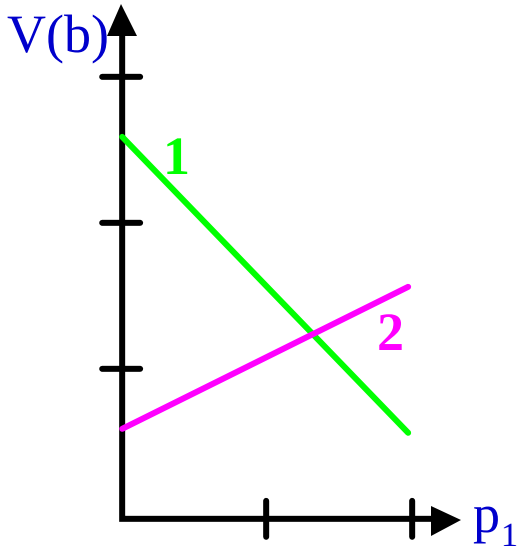
- Create new value vector using support vectors: **[1, 3], [1, 3]**

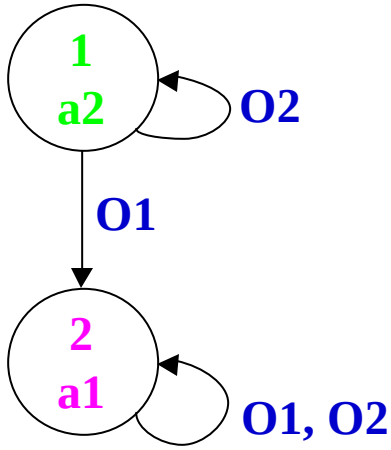
Policy Iteration for POMDPs

- ***Policy Iteration***
  - Choose a policy
  - Determine the value function, based on the current policy
  - Update the value function, based on Bellman's equation
  - Update the policy and iterate (if needed)
- ***Policy Iteration for POMDPs***
  - Original algorithm (Sondik) very inefficient and complex
  - Mainly due to evaluation of value function from policy!
  - Represent policy using finite-state controller (Hansen 1997):
    - Easy to evaluate
    - Easy to update

POMDP Policy Iteration (Hansen 1997)

- Key Idea: Represent Policy as Finite-State Controller
  - Explicitly represents: "do action then continue with given policy"
  - Nodes correspond to vectors in value function
  - Edges correspond to transitions based on observations





POMDP Policy Iteration

$$V(b) = \max_a \{R(a, b) + \gamma \sum_{b'} p(b' \mid a, b)\, V(b')\}$$

- Associate actions with initial vectors: $p(v^i) = a$

- Determine the value function, based on the current policy
  - Solve system of linear equations

$$v^i(s) = r(s, p(v^i)) + \gamma \sum_{s', o} p(o \mid p(v^i), s')\, v^{l(v^i, o)}(s')$$

- Update the value function, based on Bellman's equation
  - Can use any standard dynamic-programming method

- Update the Policy …

POMDP Policy Iteration

- Update the Policy
  - Ignore new vectors that are point-wise dominated by other vectors
  - new vectors that duplicate current vectors (same actions and observation links; point-wise equal)
  - Replace current vectors that are dominated by new vectors
  - Add new controller state otherwise