Probabilistic and Infinite Horizon Planning

4/27/2016

Massachusetts Institute of Technology

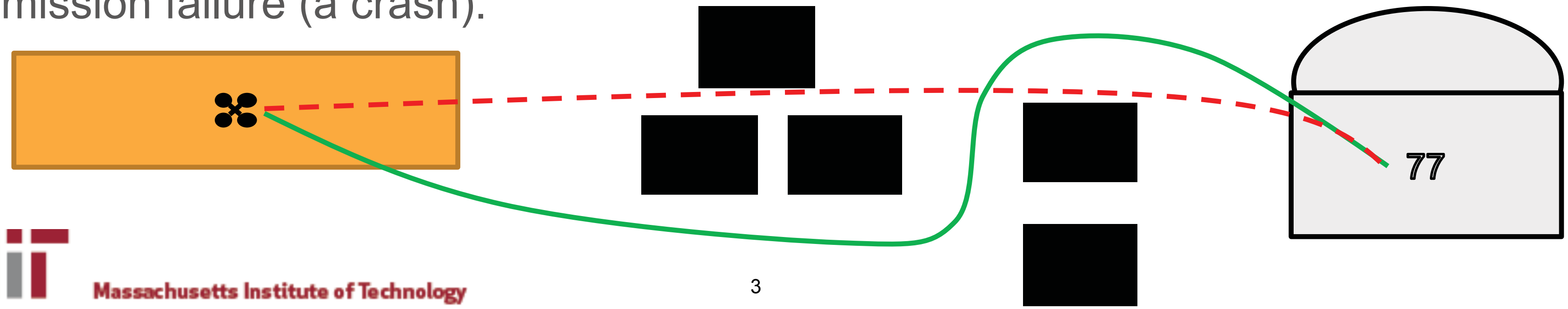
Outline

1. Quadrotor motivating example
2. Planning with Markov Processes
    1. Markov Decision Process formulation
    2. Value Iteration Algorithm
    3. Heuristic-Guided solvers
3. Extensions to Partially Observable Markov Decision Processes
    1. Partially Observable Markov Decision Process formulation
    2. PRMs in the belief space
    3. Results from FIRM case study

Motivating Example: Quadrotor Motion Planning

Given a start configuration, a goal configuration, a set of possible actions, and a cost function, find the optimal sequence of actions to get the quadrotor to the goal configuration.

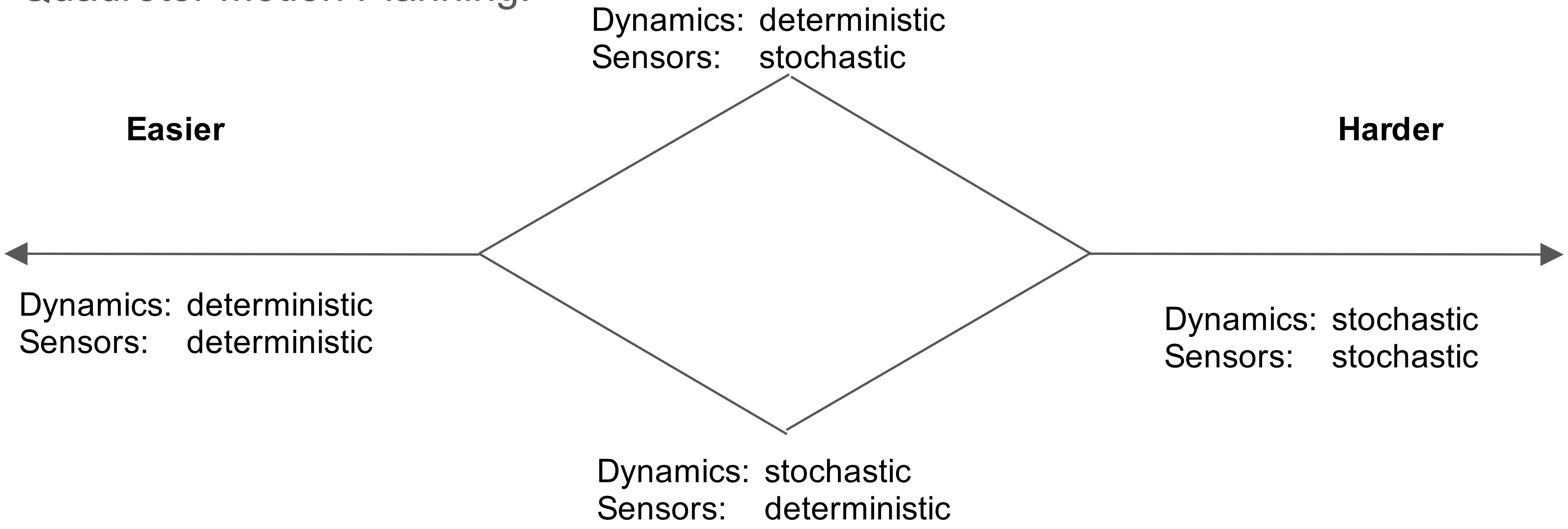
E.g. Fly a quadrotor from an Amazon fulfillment center to 77 Mass. Ave. The quadrotor may individually control its four motors, and it may take photos with an onboard camera. Minimize the expected time of the journey times the probability of mission failure (a crash).



Massachusetts Institute of Technology

Motivating Example

Quadrotor Motion Planning:



Motivating Example

Quadrotor Motion Planning:

Dynamics: deterministic
Sensors: stochastic

Easier

Harder

Dynamics: deterministic
Sensors: deterministic

Dead reckoning, validate through sensing

Dynamics: stochastic
Sensors: stochastic

Dynamics: stochastic
Sensors: deterministic

Motivating Example

Quadrotor Motion Planning:

Easier

Dynamics: deterministic
Sensors: deterministic

Dead reckoning, validate through sensing

Dynamics: deterministic
Sensors: stochastic

Dead reckoning, Kalman filtering

Dynamics: stochastic
Sensors: deterministic

Harder

Dynamics: stochastic
Sensors: stochastic

Motivating Example

Quadrotor Motion Planning:

Easier

Dynamics: deterministic
Sensors: deterministic

Dead reckoning, validate through sensing

Dynamics: deterministic
Sensors: stochastic

Dead reckoning, Kalman filtering

Dynamics: stochastic
Sensors: deterministic

Markov Decision Processes!

Harder

Dynamics: stochastic
Sensors: stochastic

Massachusetts Institute of Technology

Motivating Example

Quadrotor Motion Planning:

Easier

Harder

Dynamics: deterministic
Sensors: deterministic

Dead reckoning, validate through sensing

Dynamics: deterministic
Sensors: stochastic

Dead reckoning, Kalman filtering

Dynamics: stochastic
Sensors: deterministic

Markov Decision Processes!

Dynamics: stochastic
Sensors: stochastic

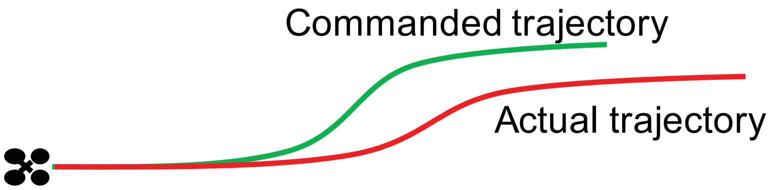
Partially Observable Markov Decision Processes!

Motivating Example

Quadrotor Motion Planning

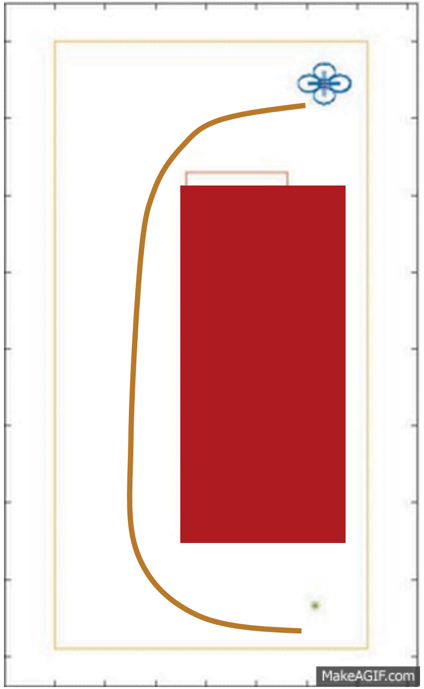
Action Uncertainty:

1) Noisy motors
2) Wind gusts
3) Dropped command signals



Probabilistic Modelling

Modelling stochastic processes deterministically is not only inaccurate, it can be dangerous



Massachusetts Institute of Technology

Outline

1. Quadrotor motivating example
2. Planning with Markov Processes
    1. Markov Decision Process formulation
    2. Value Iteration Algorithm
    3. Heuristic-Guided solvers
3. Extensions to Partially Observable Markov Decision Processes
    1. Partially Observable Markov Decision Process formulation
    2. PRMs in the belief space
    3. Results from FIRM case study

Markov Decision Processes (MDPs)

A Markov decision process is defined as a tuple with five elements:

S: a finite set of states
A: a finite set of actions
$P(s'|s,a)$: a transition model expressing the probability of reaching state s' from s after taking action a
$R(s,a,s')$: an immediate reward for moving from state s to s' by using action a
γ: a discount factor for rewards

We want to find a policy π that generates the optimal action from each state

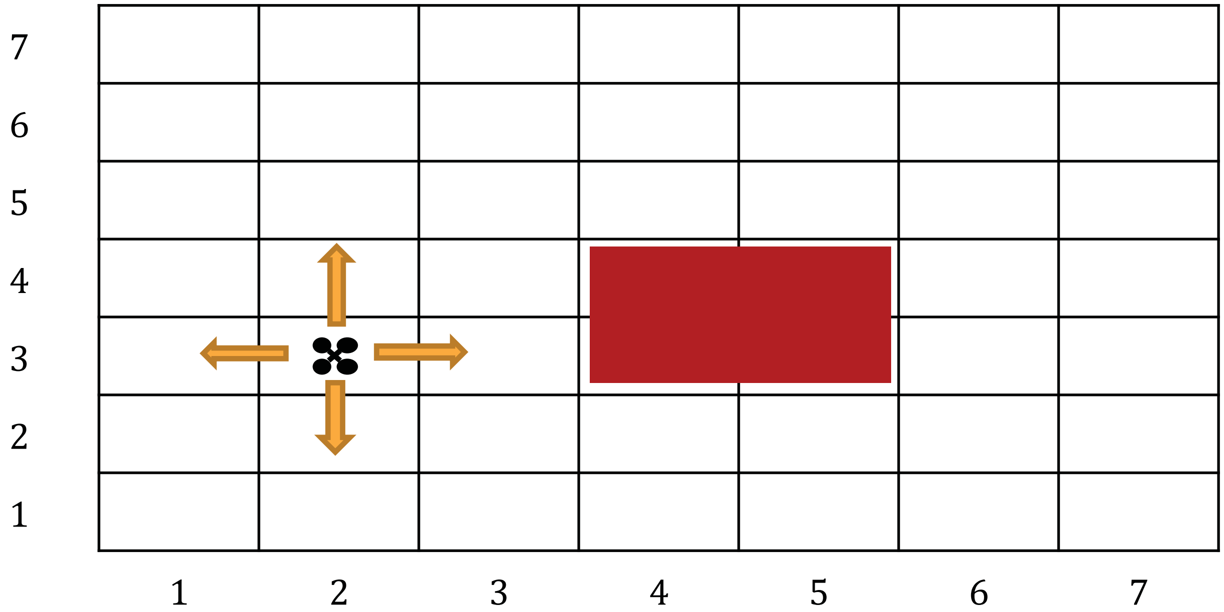
maximize the expected lifetime reward:

$$\mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$$

Massachusetts Institute of Technology

MDP Example

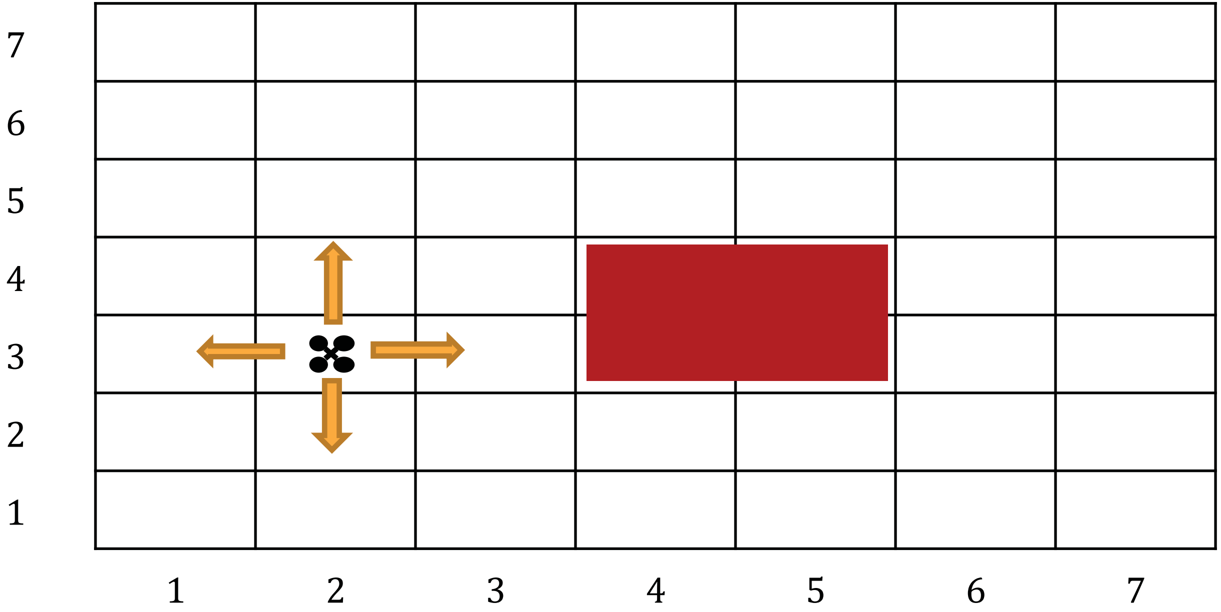
Quadrotor with perfect sensor



$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$

MDP Example

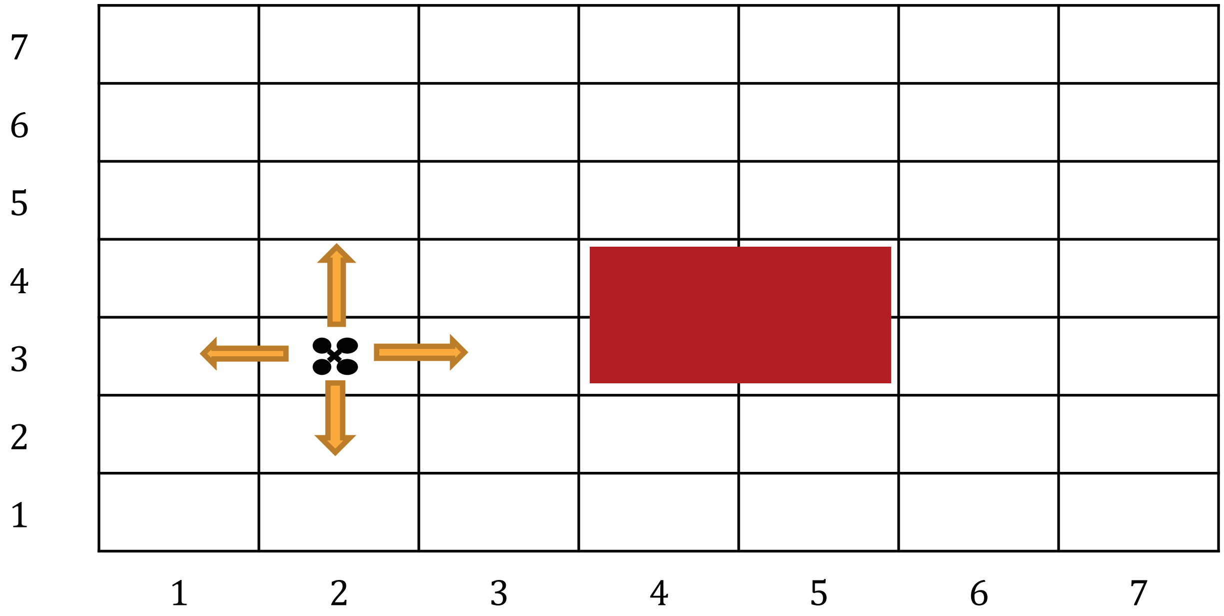
Quadrotor with perfect sensor



$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

MDP Example

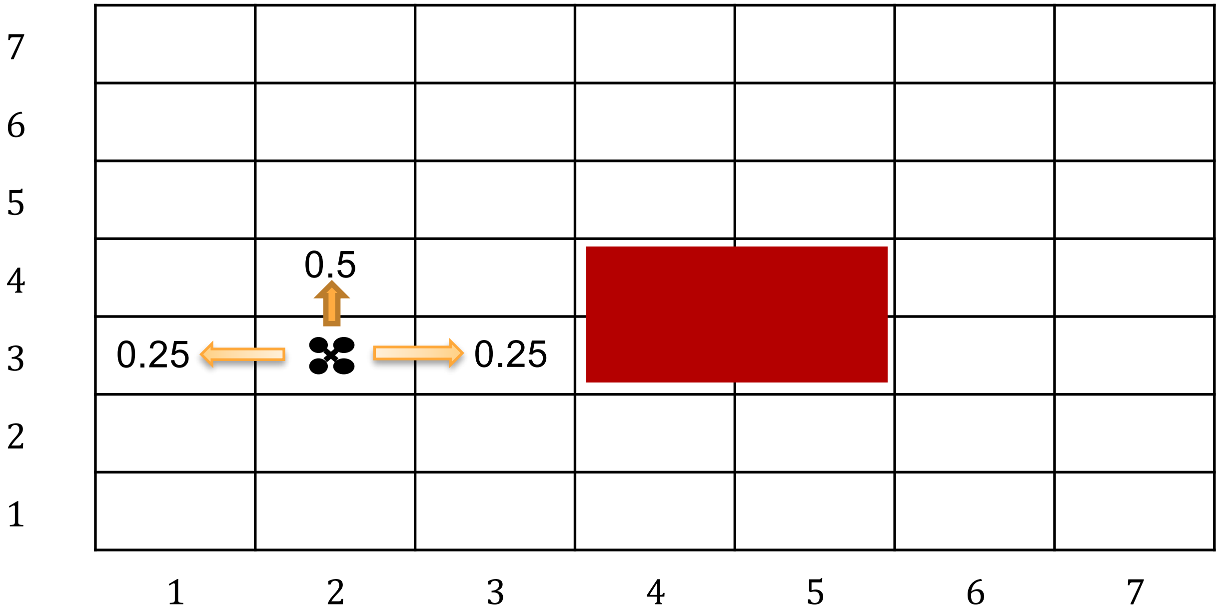
Quadrotor with perfect sensor



$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$
$$A = \{N, S, E, W, null\}$$
$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

MDP Example

Quadrotor with perfect sensor



MDP Example

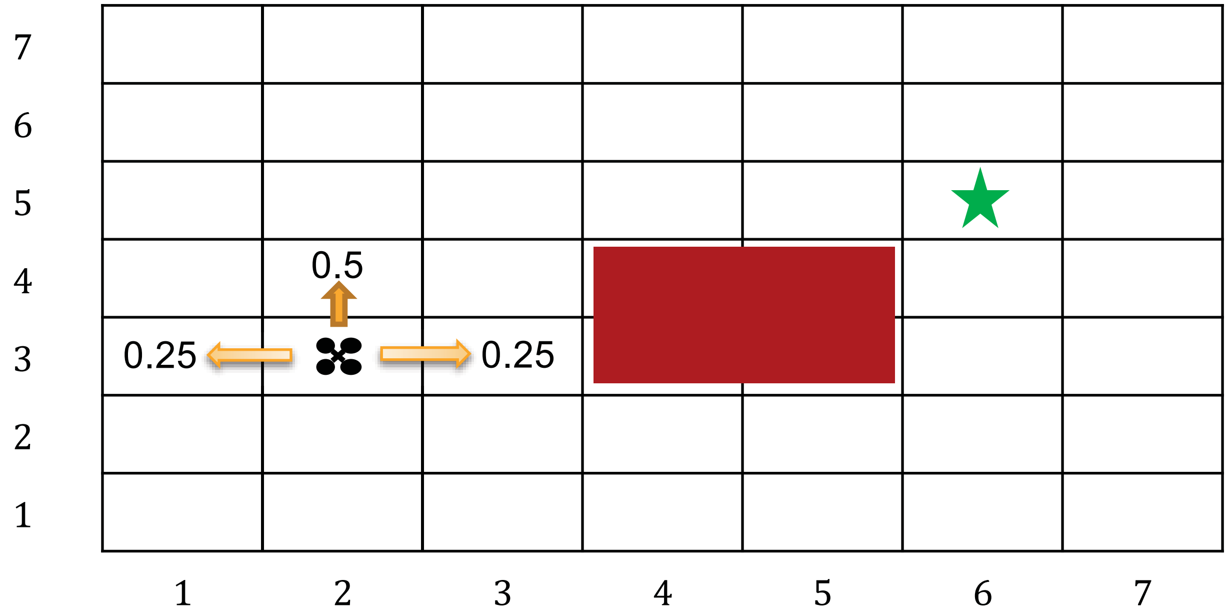
Quadrotor with perfect sensor

$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$

$$A = \{N, S, E, W, null\}$$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$



MDP Example

Quadrotor with perfect sensor

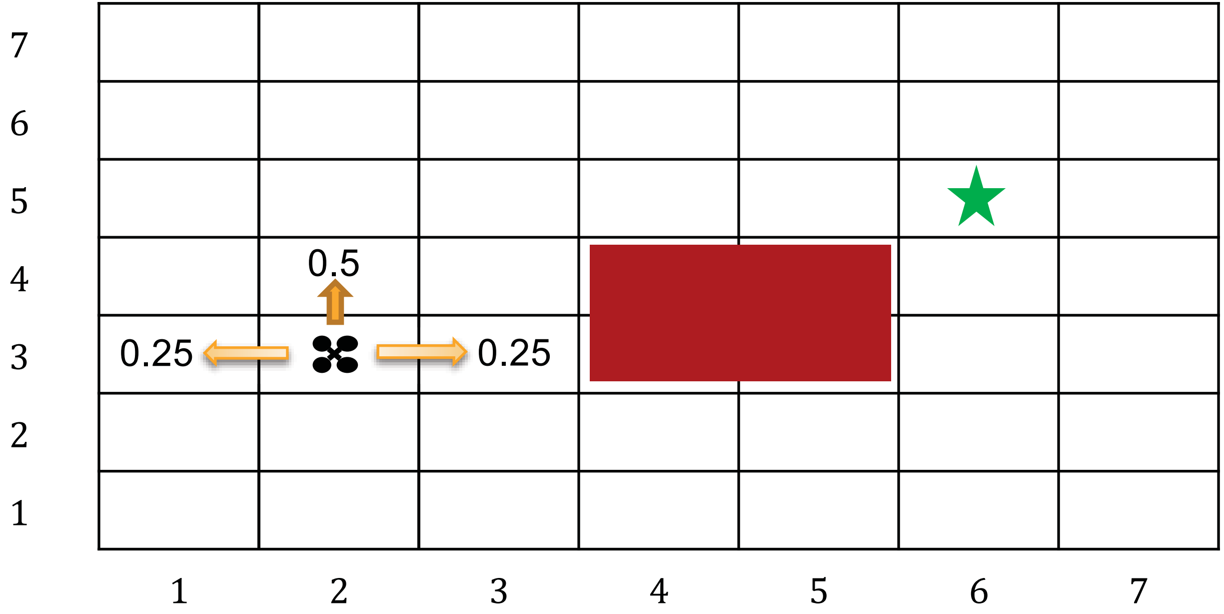
$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

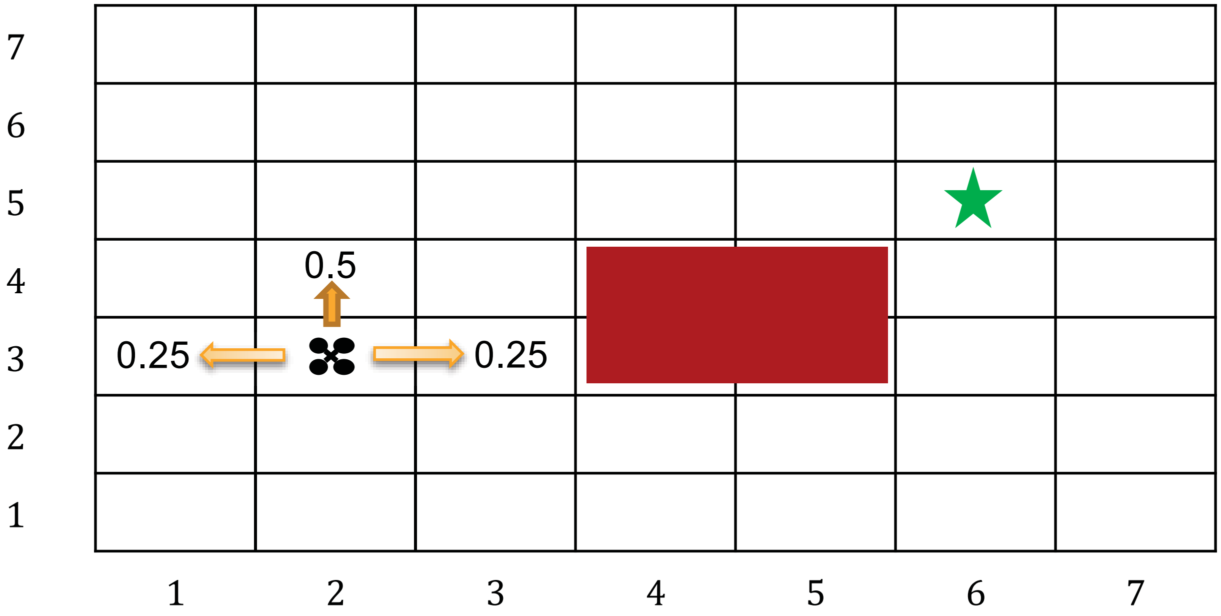
$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$



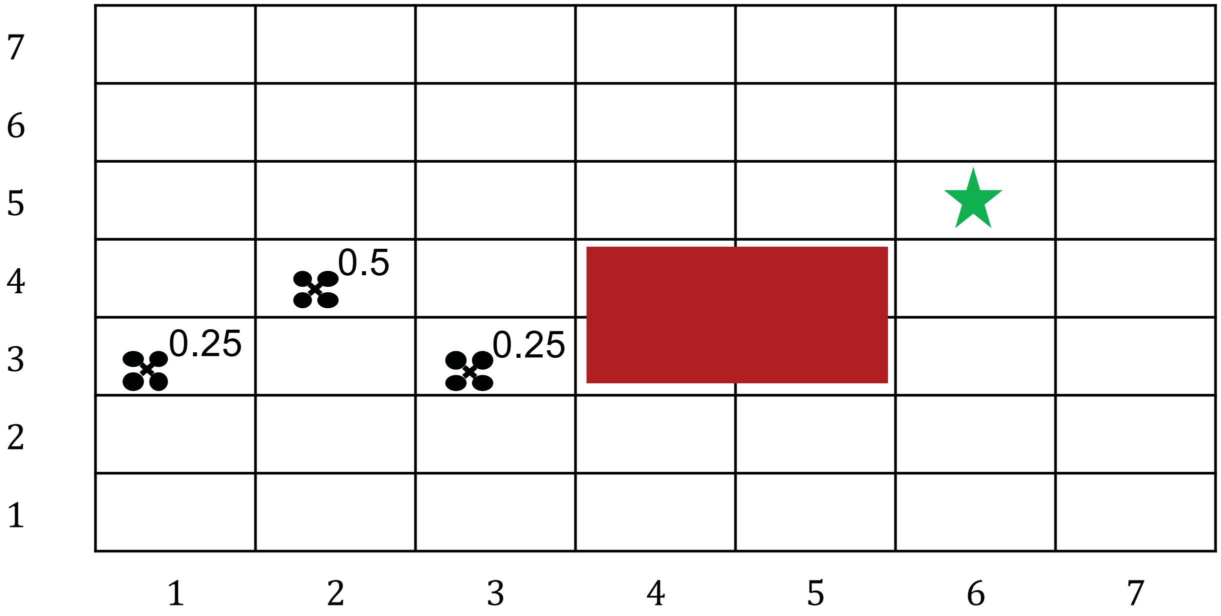
MDP Example

$$a_1 = N$$



MDP Example

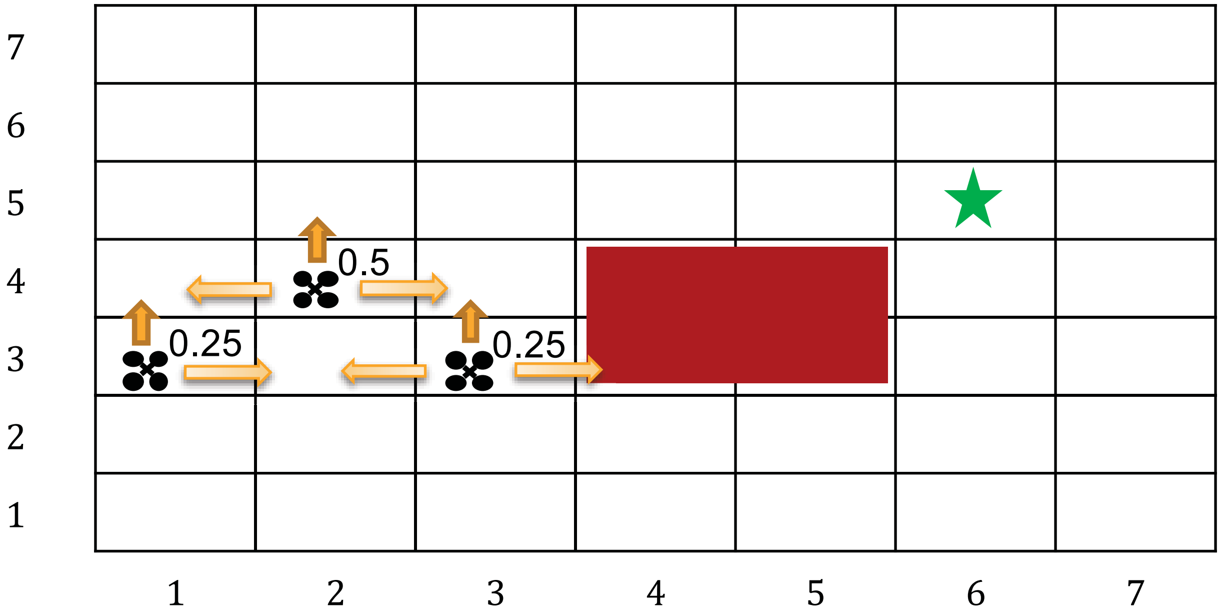
$$a_1 = N$$



Massachusetts Institute of Technology

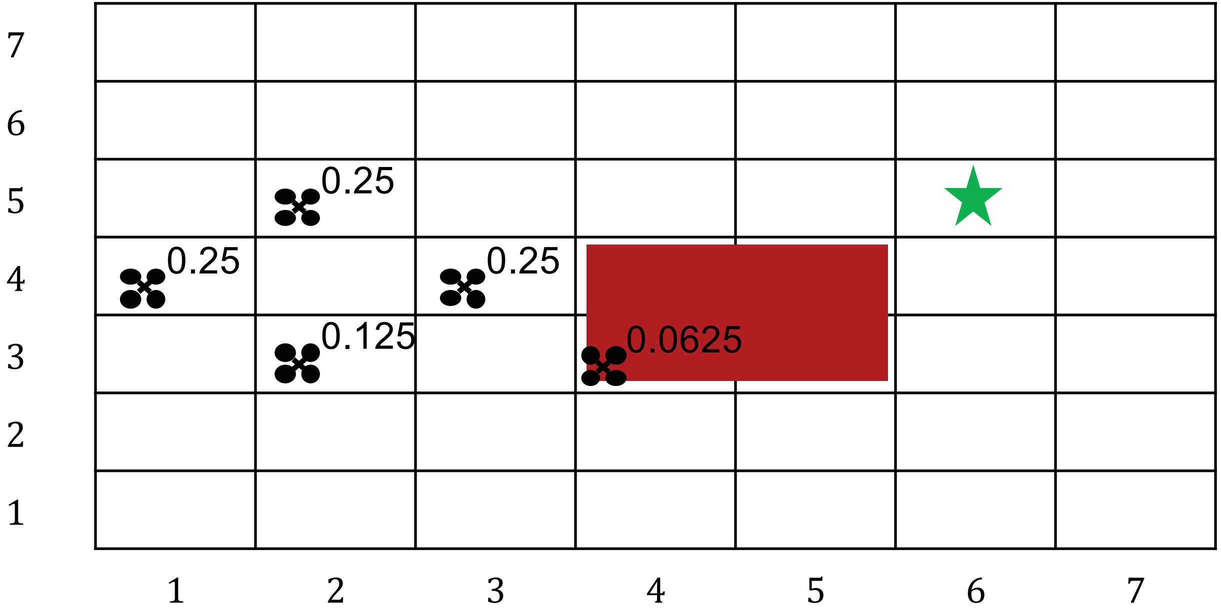
MDP Example

$$a_1 = N, a_2 = N$$



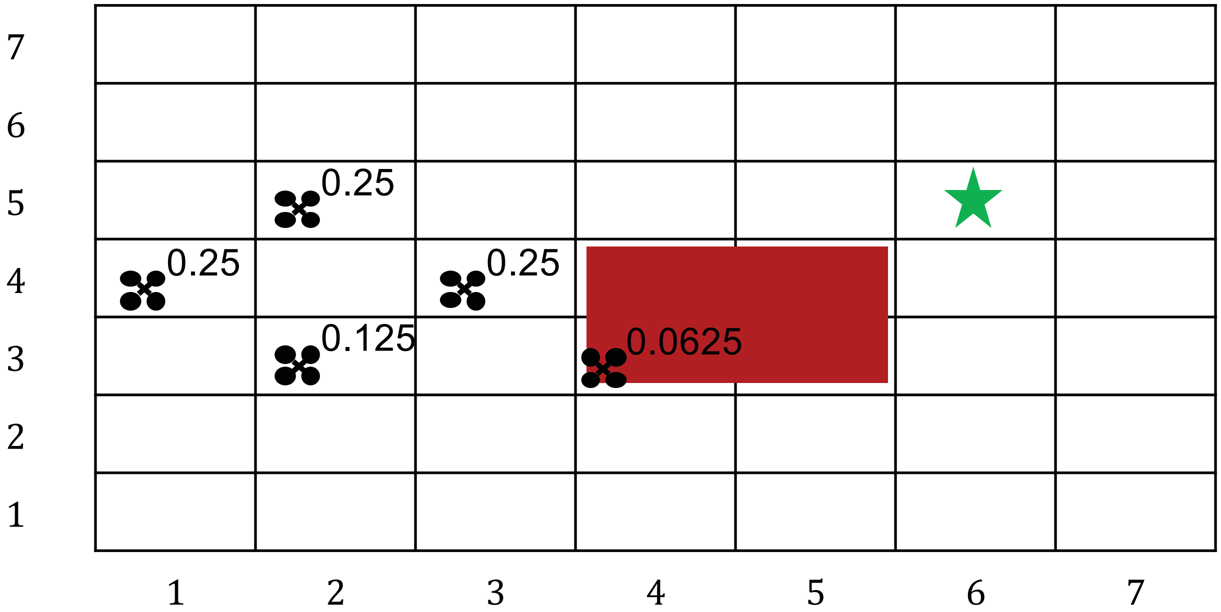
MDP Example

$$a_1 = N, a_2 = N$$



MDP Example

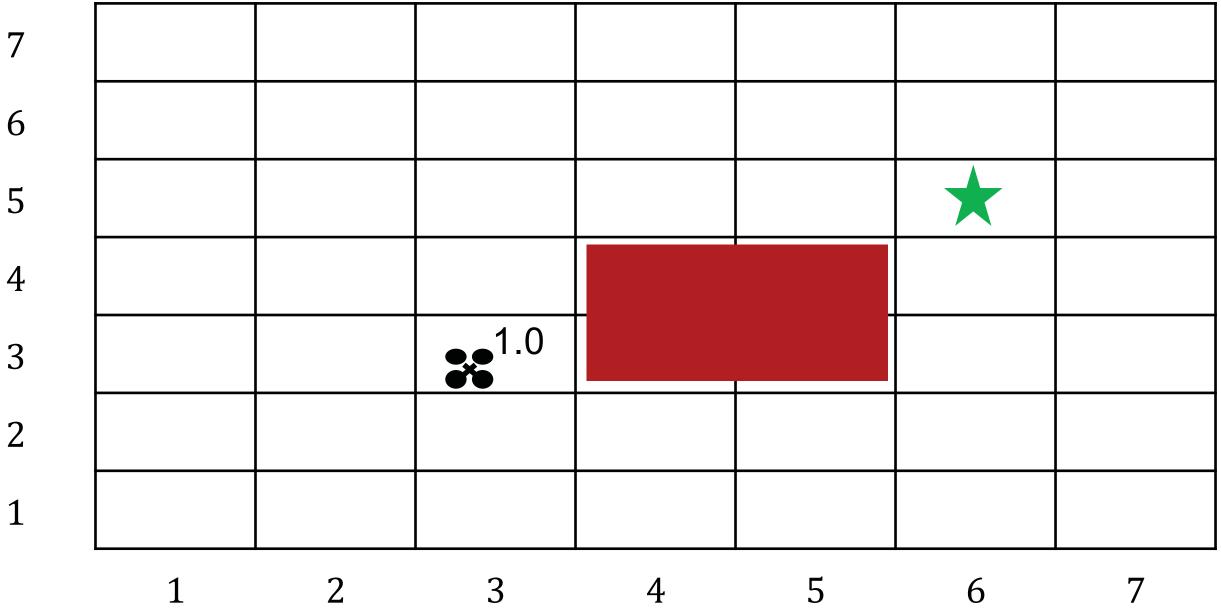
How do we collapse this distribution?



Massachusetts Institute of Technology

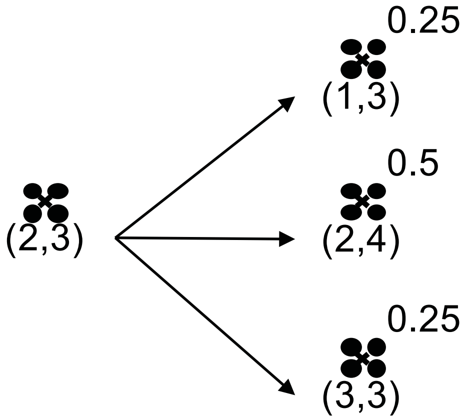
MDP Example

How do we collapse this distribution? Observations: $o_1 = (3,3)$

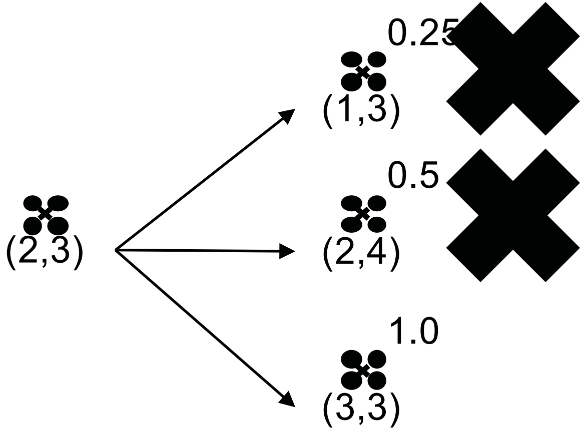


MDP Example: Tree view

$a_1 = N$



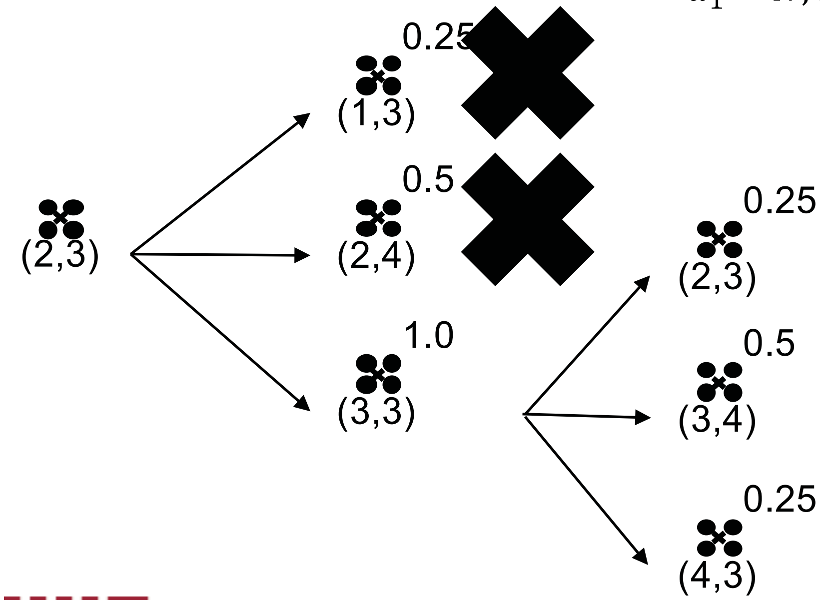
MDP Example: Tree view



$a_1 = N, o_1 = (3,3)$

MDP Example: Tree view

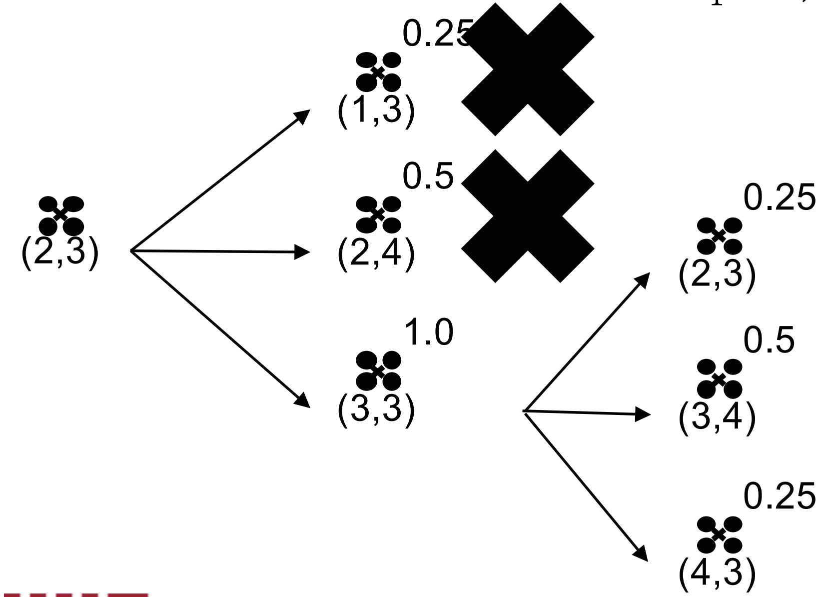
$a_1 = N, o_1 = (3,3), a_2 = N$



Massachusetts Institute of Technology

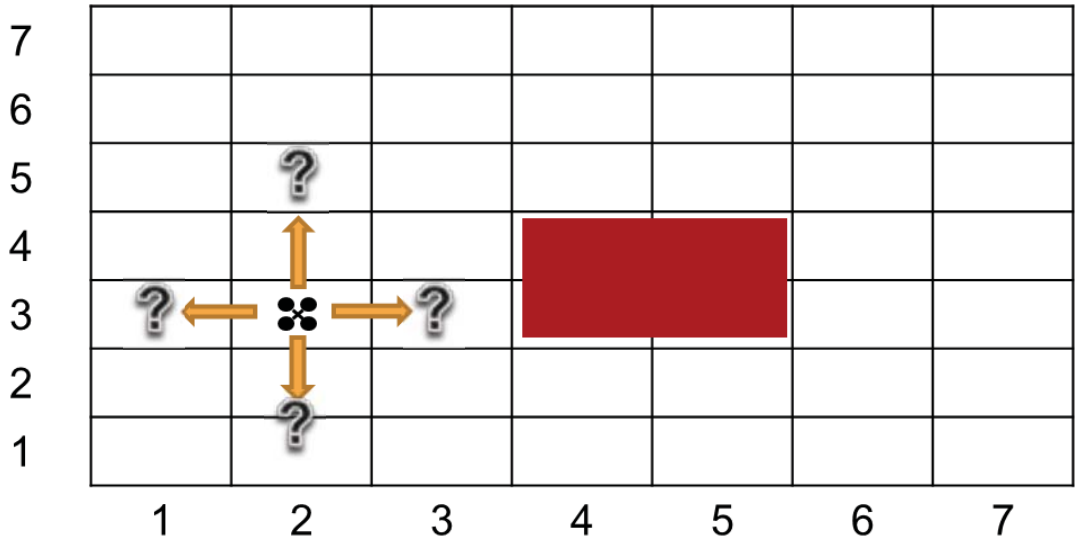
MDP Example: Tree view

$a_1 = N, o_1 = (3,3), a_2 = N$



We can completely collapse the distribution every time we make an observation

How do we find the optimal policy?



Dynamic Programming:
Value Iteration
Policy Iteration

Outline

1. Quadrotor motivating example
2. Planning with Markov Processes
    1. Markov Decision Process formulation
    2. Value Iteration Algorithm
    3. Heuristic-Guided solvers
3. Extensions to Partially Observable Markov Decision Processes
    1. Partially Observable Markov Decision Process formulation
    2. PRMs in the belief space
    3. Results from FIRM case study

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

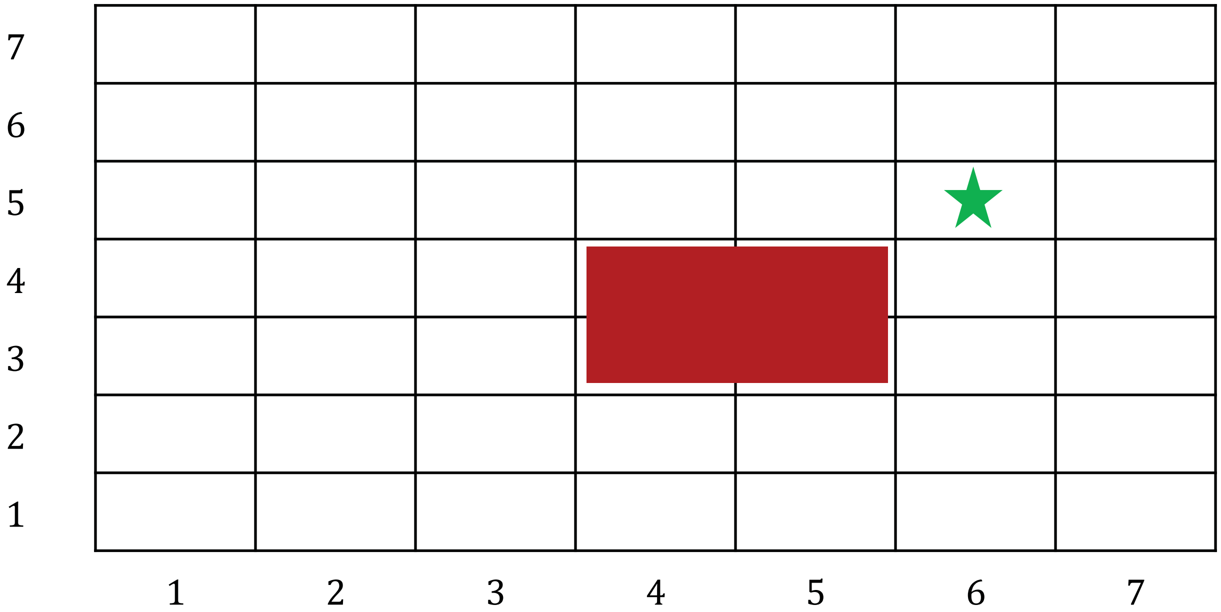
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

maximize expected reward sum: $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$



Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

maximize expected reward sum: $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$

Initialize value at each state to zero

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Value Iteration

t_0
$s = (6,5)$
$a = null$

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Massachusetts Institute of Technology

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

t_0
$s = (6,5)$
$a = null$
$P((6,5)|(6,5),null) = .5$
$P((5,5)|(6,5),null) = .25$
$P((7,5)|(6,5),null) = .25$

	1	2	3	4	5	6	7
7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

Value Iteration

t_0

$s = (6,5)$

$a = null$

$P((6,5)|(6,5), null) = .5$

$P((5,5)|(6,5), null) = .25$

$P((7,5)|(6,5), null) = .25$

$V_0(6,5) = 0$

$V_0(5,5) = 0$

$V_0(7,5) = 0$

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

t_0

$s = (6,5)$

$a = null$

$P((6,5)|(6,5), null) = .5$

$P((5,5)|(6,5), null) = .25$

$P((7,5)|(6,5), null) = .25$

$V_0(6,5) = 0$

$V_0(5,5) = 0$

$V_0(7,5) = 0$

$R((6,5), null) = 1$

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Massachusetts Institute of Technology

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

t_0
$s = (6,5)$
$a = null$
$P((6,5)|(6,5),null) = .5$
$P((5,5)|(6,5),null) = .25$
$P((7,5)|(6,5),null) = .25$
$V_0(6,5) = 0$
$V_0(5,5) = 0$
$V_0(7,5) = 0$
$R((6,5),null) = 1$

$\boldsymbol{V_1(6,5)}$:

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

t_0
$s = (6,5)$
$a = null$
$P((6,5)|(6,5),null) = .5$
$P((5,5)|(6,5),null) = .25$
$P((7,5)|(6,5),null) = .25$
$V_0(6,5) = 0$
$V_0(5,5) = 0$
$V_0(7,5) = 0$
$R\big((6,5),null,s'\big) = 1$

$\boldsymbol{V_1(6,5)}$:

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

$$(.5(\qquad\qquad)) + (.25(\qquad\qquad)) + (.25(\qquad\qquad))$$

$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$
$$A = \{N, S, E, W, null\}$$
$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$
$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$
$$\gamma = 0.9$$

Value Iteration

t_0
$s = (6,5)$
$a = null$
$P((6,5)|(6,5), null) = .5$
$P((5,5)|(6,5), null) = .25$
$P((7,5)|(6,5), null) = .25$
$V_0(6,5) = 0$
$V_0(5,5) = 0$
$V_0(7,5) = 0$
$R((6,5), null) = 1$

$\boldsymbol{V_1(6,5)}$:

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

$$\mathbf{(.5(1 \qquad)) + (.25(1 \qquad)) + (.25(1 \qquad))}$$

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

Value Iteration

t_0
$s = (6,5)$
$a = null$
$P((6,5)|(6,5), null) = .5$
$P((5,5)|(6,5), null) = .25$
$P((7,5)|(6,5), null) = .25$
$V_0(6,5) = 0$
$V_0(5,5) = 0$
$V_0(7,5) = 0$
$R\big((6,5), null, s'\big) = 1$

$\boldsymbol{V_1(6,5)}$:

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

$$\boldsymbol{(.5(1 + .9 * 0)) + (.25(1 + .9 * 0)) + (.25(1 + .9 * 0))}$$

Value Iteration

$$\boldsymbol{V_1(6,5)}: (.5(1 + .9 * 0)) + (.25(1 + .9 * 0)) + (.25(1 + .9 * 0))$$

$$V_{t+1}(s): \sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s'))$$

$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$

$$A = \{N, S, E, W, null\}$$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$$\gamma = 0.9$$

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Value Iteration

What about the other actions?

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
4	0	0	0			0	0
3	0	0	0			0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Value Iteration

$$V_{t+1}(s): \sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s'))$$

Repeat over all actions and take maximum

$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$

$$A = \{N, S, E, W, null\}$$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$$\gamma = 0.9$$

	5	6	7
7	0	0	0
6	0	0	0
5	0	0	0
4			0
3		0	0
2	0	0	0
1	0	0	0

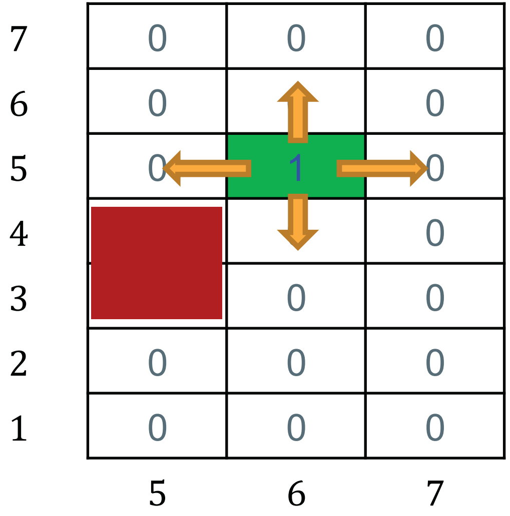
$$S = \{1,2,3,4,5,6,7\}\times\{1,2,3,4,5,6,7\}$$
$$A = \{N, S, E, W, null\}$$
$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$
$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$
$$\gamma = 0.9$$

Value Iteration

$$V_{t+1}(s): \sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s'))$$

Repeat over all actions and take maximum

$$V_1(6,5) = \max_{a \in A(6,5)} \begin{bmatrix} N: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ S: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ E: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ W: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ null: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \end{bmatrix} = 1$$



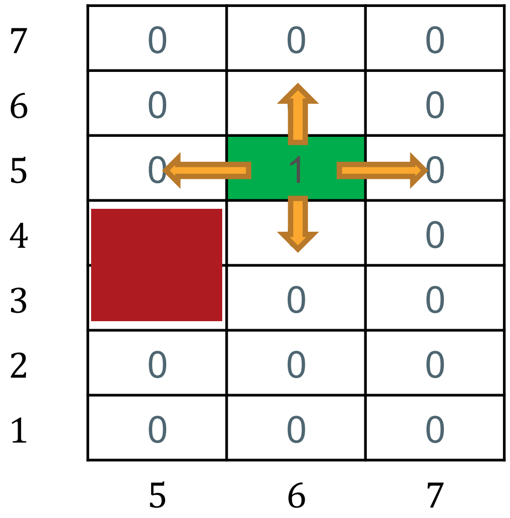
Value Iteration

$$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$$
$$A = \{N, S, E, W, null\}$$
$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$
$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$
$$\gamma = 0.9$$

$$V_{t+1}(s): \sum_{s' \in S} p_{ss'}(s'|s,a)\left(R(s,a,s') + \gamma V_t(s')\right)$$

Repeat over all actions and take maximum

$$V_1(6,5) = \max_{a \in A(6,5)} \begin{bmatrix} N: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ S: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ E: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ W: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \\ null: [(.5(1+.9*0)) + (.25(1+.9*0)) + (.25(1+.9*0))] \end{bmatrix} = 1$$



Bellman back-up equation:

$$V_{t+1}(s) = \max_{a \in A(s)} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)\left(R(s,a,s') + \gamma V_t(s')\right)\right]$$

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

Bellman back-up equation:

$$V_{t+1}(s) = \max_{a \in A(s)} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s')) \right]$$

Iteratively calculate values across entire state space

$t = 0$

	5	6	7
7	0	0	0
6	0	0	0
5	0	0	0
4		0	0
3		0	0
2	0	0	0
1	0	0	0

Massachusetts Institute of Technology

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

Bellman back-up equation:

$$V_{t+1}(s) = \max_{a \in A(s)} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s')) \right]$$

Iteratively calculate values across entire state space

$t = 1$

7	0	0	0
6	0	0	0
5	0	1	0
4		0	0
3		0	0
2	0	0	0
1	0	0	0
	5	6	7

Value Iteration

$S = \{1,2,3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$
$A = \{N, S, E, W, null\}$

$$P(s'|s,a) = \begin{cases} 0.50 & \text{if } s' \text{ along } a \\ 0.25 & \text{if } s' \text{ right of } a \\ 0.25 & \text{if } s' \text{ left of } a \end{cases}$$

$$R(s,a,s') = \begin{cases} 0 & \text{if } s \neq (6,5) \\ 1 & \text{if } s = (6,5) \end{cases}$$

$\gamma = 0.9$

Bellman back-up equation:

$$V_{t+1}(s) = \max_{a \in A(s)} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)\left(R(s,a,s') + \gamma V_t(s')\right) \right]$$

	5	6	7
7	0	0	0
6	.225	.45	.225
5	.45	1.9	.45
4		.45	.225
3		0	0
2	0	0	0
1	0	0	0

Iteratively calculate values across entire state space

$t = 2$

Value Iteration

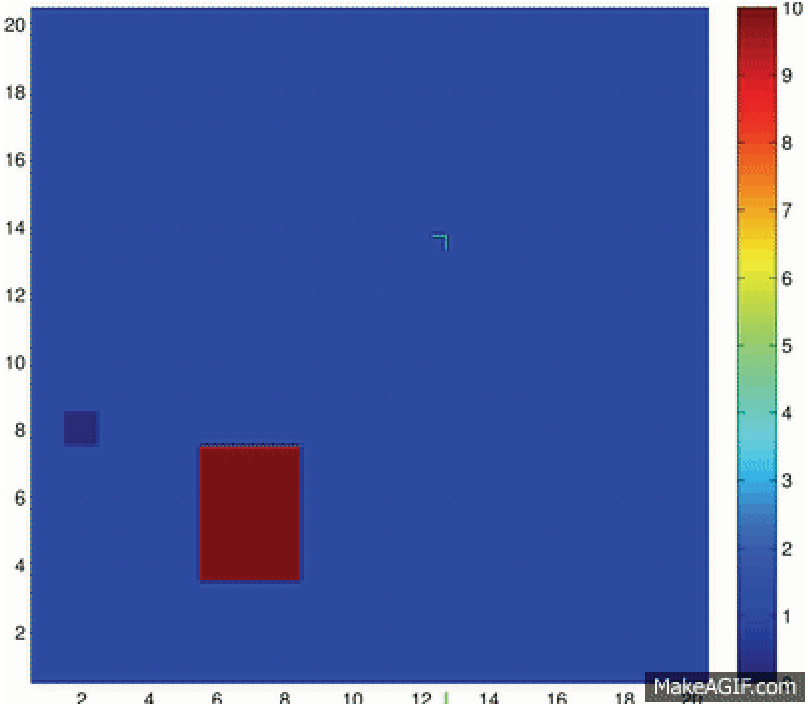
Terminate when change in values is small, making our approximations “close enough”

$$\|V_{t+1} - V_t\| < \varepsilon$$

Extract the optimal policy from lifetime values:

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)\left(R(s,a,s') + \gamma V_t(s')\right) \right]$$

Value Iteration



Value Iteration

Terminate when change in values is small, making our approximations “close enough”

$$\|V_{t+1} - V_t\| < \varepsilon$$

Extract the optimal policy from lifetime values:

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left[\sum_{s' \in S} p_{ss'}(s'|s,a)(R(s,a,s') + \gamma V_t(s')) \right]$$

Complexity for one iteration:

$$O(S^2 A)$$

Outline

1. Quadrotor motivating example
2. Planning with Markov Processes
    1. Markov Decision Process formulation
    2. Value Iteration Algorithm
    3. Heuristic-Guided solvers
3. Extensions to Partially Observable Markov Decision Processes
    1. Partially Observable Markov Decision Process formulation
    2. PRMs in the belief space
    3. Results from FIRM case study

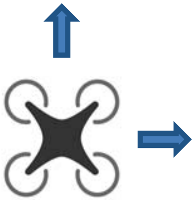
Value Iteration is SLOW!

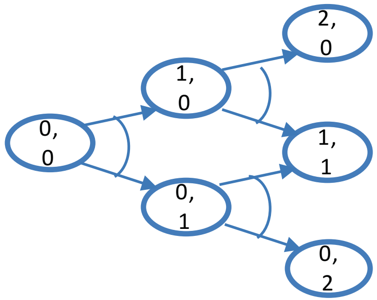
- State spaces can be very large and high-dimensional
- We don’t want to iterate over the entire space if we aren't ever going to go there
- What if we could exclude “inferior” regions from the search?
- **What algorithms perform a “best”-first search and how do they do it?**

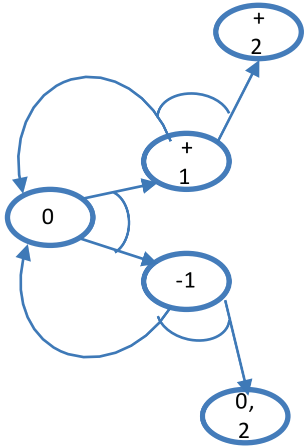
Heuristic Guided Algorithms

- A* - for deterministic graph search
- AO* - for graphs with “and” coupling
- LAO* - for AO graphs with loops
  - Most general algorithm that can deal with probabilistic coupling and revisiting states on an infinite time horizon

MDPs → AO Graphs

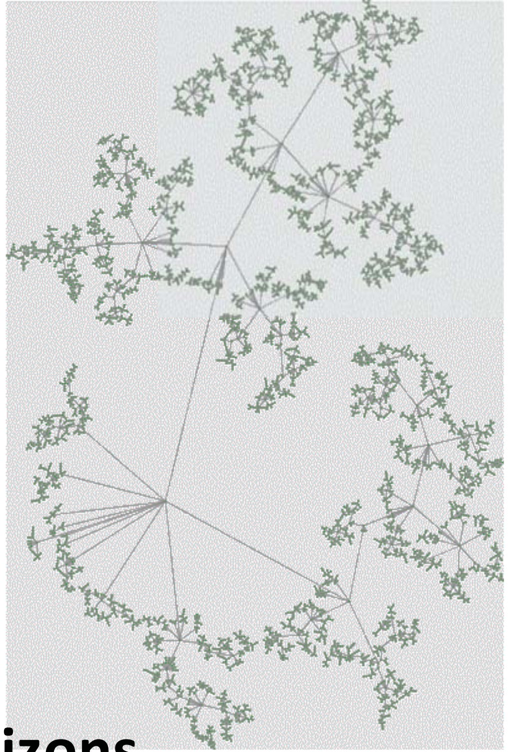






LAO*

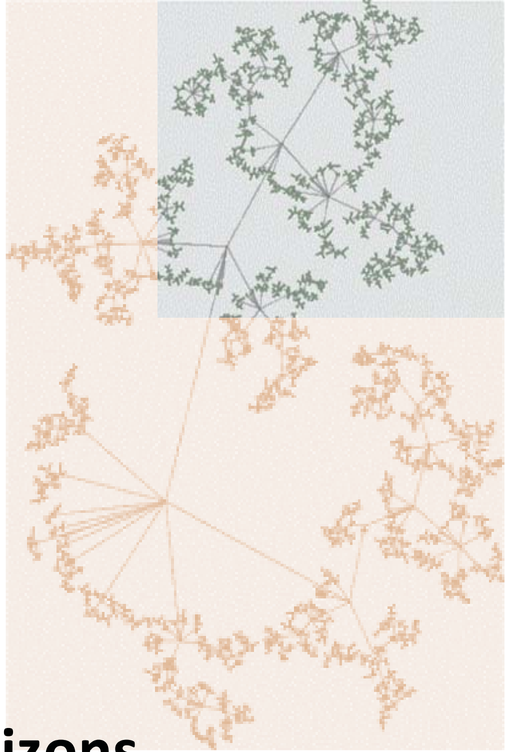
- **Heuristic guided envelope**
  - To keep problem smaller
  - Admissible
    - Underestimate costs
    - Overestimate rewards
- **Local optimal policy search**
- **Arbitrary graphs and time horizons**



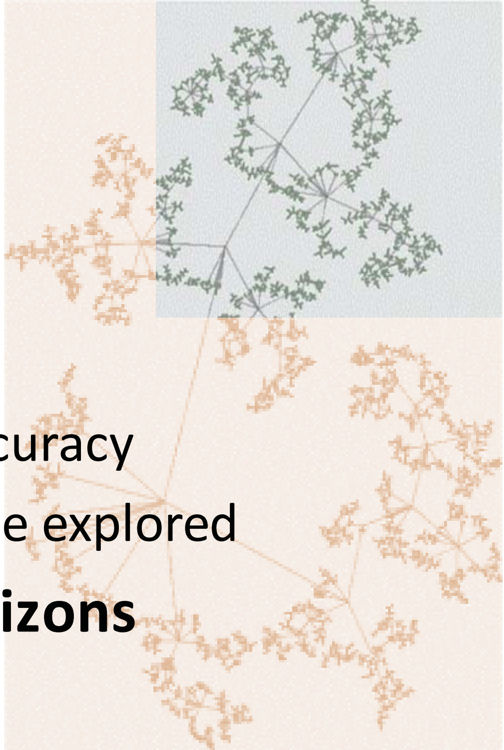
Massachusetts Institute of Technology

LAO*

- **Heuristic guided envelope**
  - To keep problem smaller
  - Admissible
    - Underestimate costs
    - Overestimate rewards
- **Local optimal policy search**
- **Arbitrary graphs and time horizons**

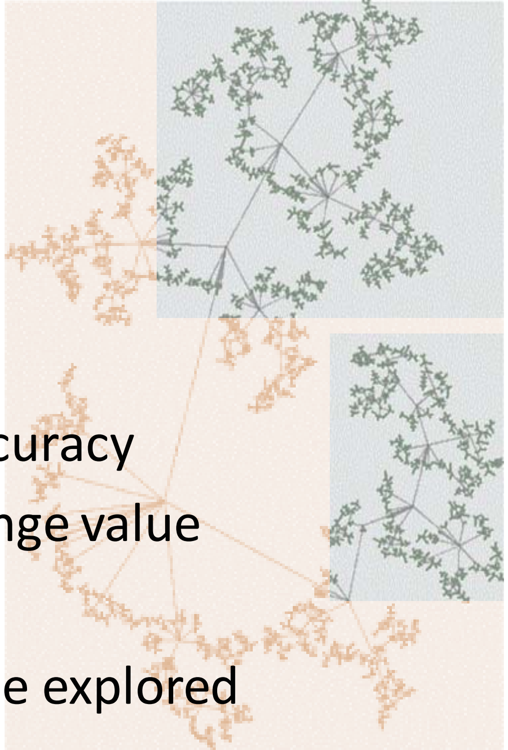


LAO*



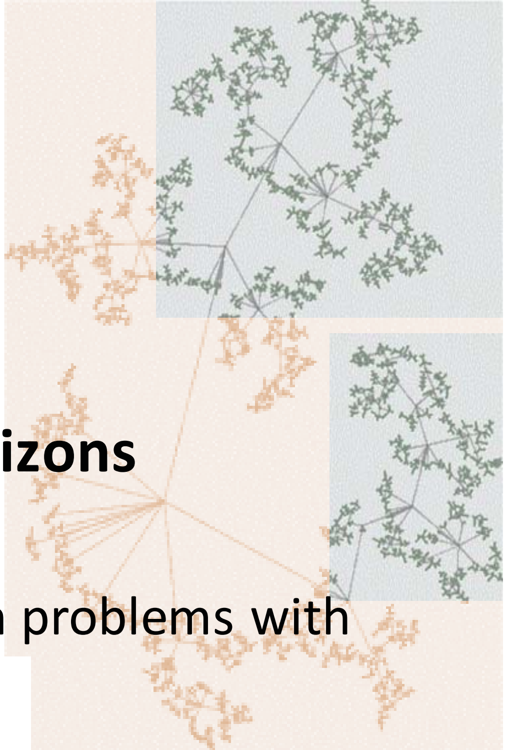
- **Heuristic guided envelope**
- **Local optimal policy search**
  - Larger policy search → more accuracy
  - Only part of state space has to be explored
- **Arbitrary graphs and time horizons**

LAO*

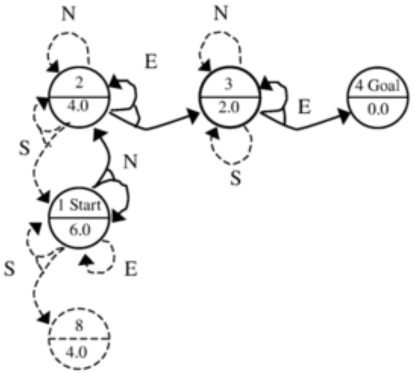


- **Heuristic guided envelope**
- **Local optimal policy search**
  - Larger policy search → more accuracy
  - Suboptimal states wouldn't change value iteration score
  - Only part of state space has to be explored
- **Arbitrary graphs and time horizons**

LAO*



- **Heuristic guided envelope**
- **Local optimal policy search**
- **Arbitrary graphs and time horizons**
  - “L” stands for loops
  - LAO* can handle infinite horizon problems with loops



Input:

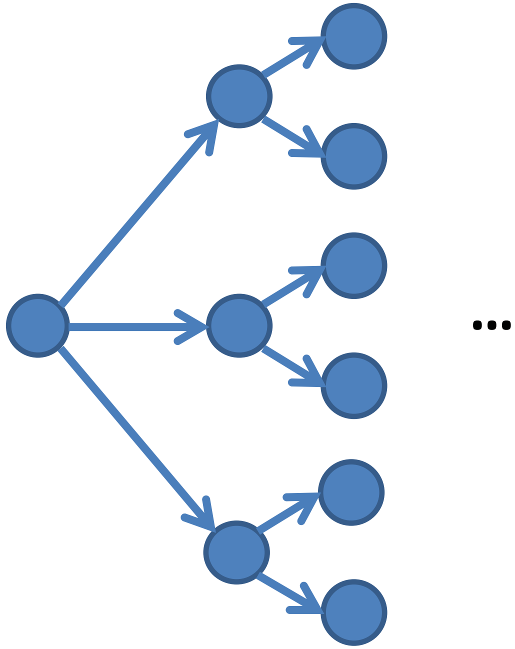
MDP or AO-Graph with:

Transition Probabilities
Reward function
Heuristic

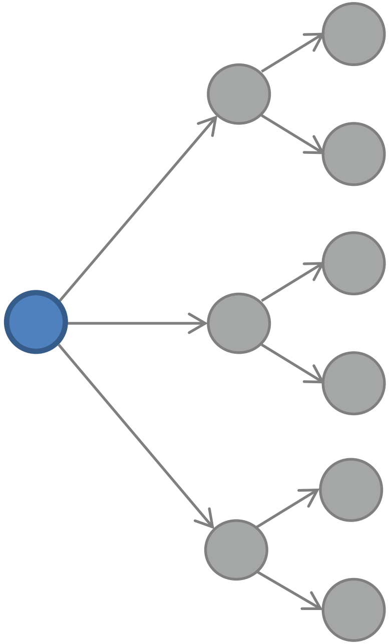
Output:

Optimal Policy for every
“reachable” state

Infinite horizon plan



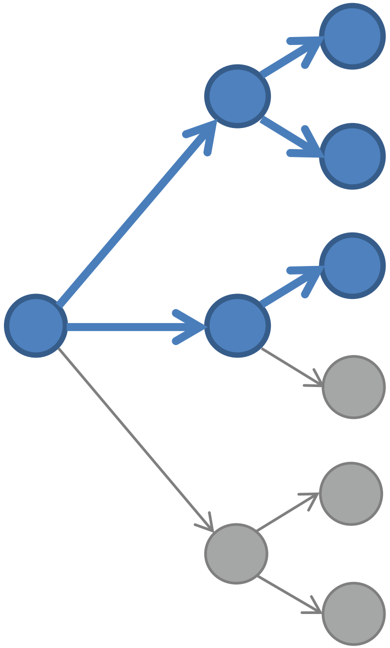
This is our State Space (S)



This is our Envelope ($S_E \subset S$)

$$S_E = \{S_0\}$$

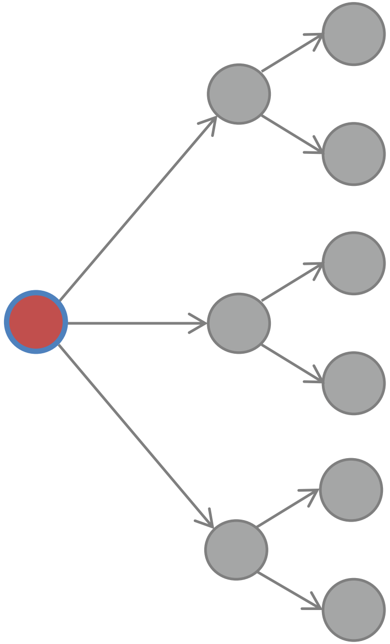
Massachusetts Institute of Technology



This is our Envelope

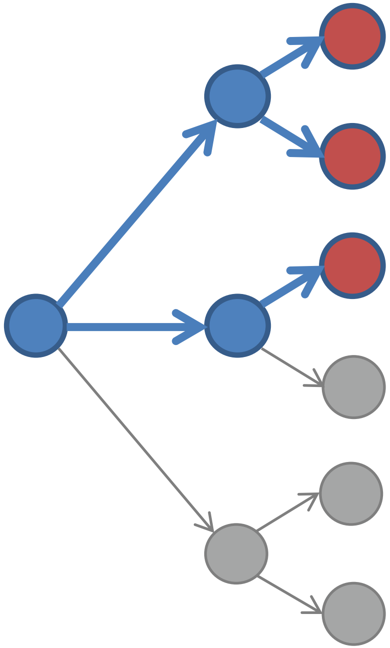
$(S_E \subset S)$

$$S_E = \{S_0\}$$



This is also our current set of terminal states ($S_T \subset S_E \subset S$)

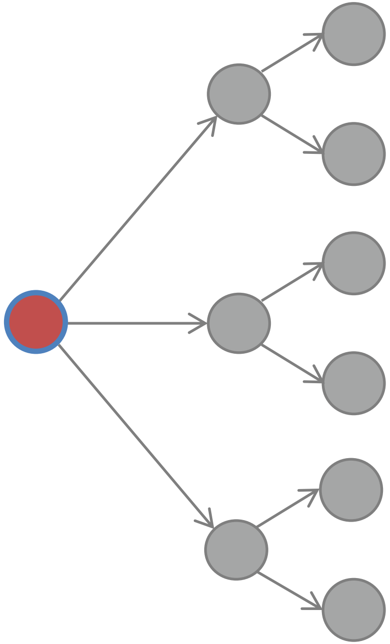
$$S_T = \{S_0\}$$



This is also our current set of terminal states

$(S_T \subset S_E \subset S)$

$$S_T = \{S_0\}$$



Optimal Policy Search

S_E States in Envelope

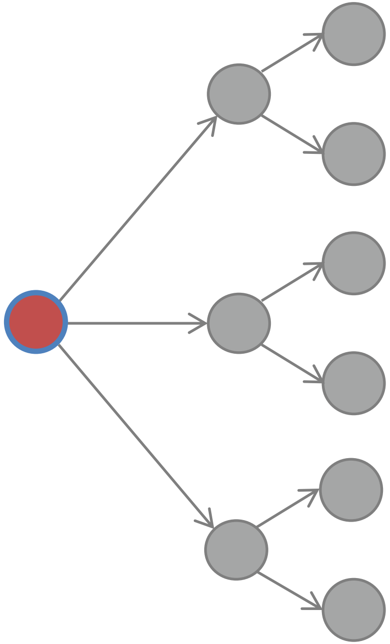
R_E Costs or Heuristics

T_E Transition Probabilities

$$T_E(s'|s,a) = \begin{cases} 0 & \text{if } S \in S_T \\ \Pr(s'|s,a) & \text{otherwise} \end{cases}$$

$$R_E(s,a) = \begin{cases} h(s) & \text{if } S \in S_T \\ \mathrm{R}(s,a) & \text{otherwise} \end{cases}$$

Massachusetts Institute of Technology



Optimal Policy Search

$$\pi \ :: < S_E, R_E, T_E >$$

Search for optimal policy on entire envelope

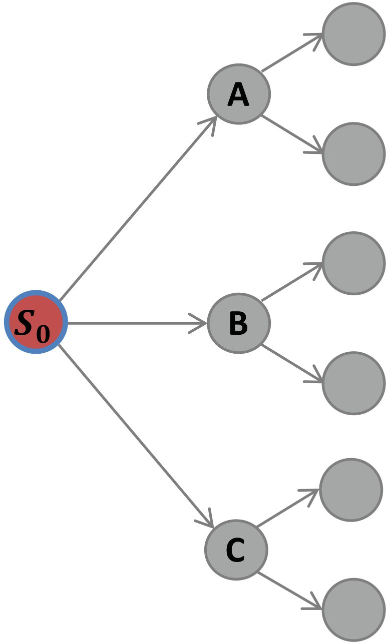
Using Value Iteration

Massachusetts Institute of Technology

LAO* Steps

1. Create R_E and T_E
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as ${S_T}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set

   (as long as they weren't in the envelope before)
5. Add the new children to the envelope
6. Repeat until no states are expanded



1. **Create R_E and T_E**
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

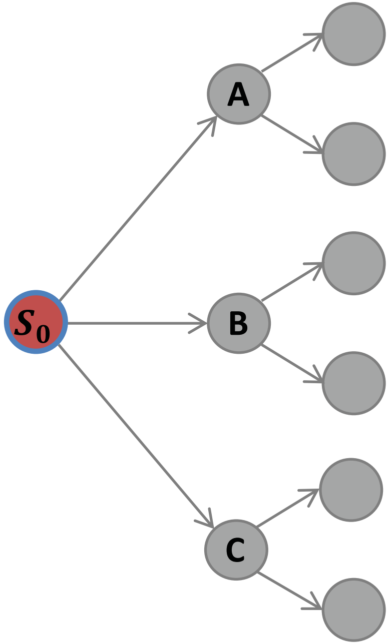
$$T_E(s'|s,a) = \begin{cases} 0 & \text{if } S \in S_T \\ \Pr(s'|s,a) & \text{otherwise} \end{cases}$$

$$R_E(s,a) = \begin{cases} h(s) & \text{if } S \in S_T \\ \mathrm{R}(s,a) & \text{otherwise} \end{cases}$$

$$\boldsymbol{R_E(s,a) = \{S_0 : 20\}}$$

$$\boldsymbol{T_E(s'|s,a) = \{S_0 \to \{A, B, C\} : 0\}}$$

Massachusetts Institute of Technology



1. Create R_E and T_E
2. **Find optimal policy on envelope**
3. Select the all the terminal states that are reachable by the optimal policy as S_T^{π}
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

$$\pi :: < S_E, R_E, T_E >$$

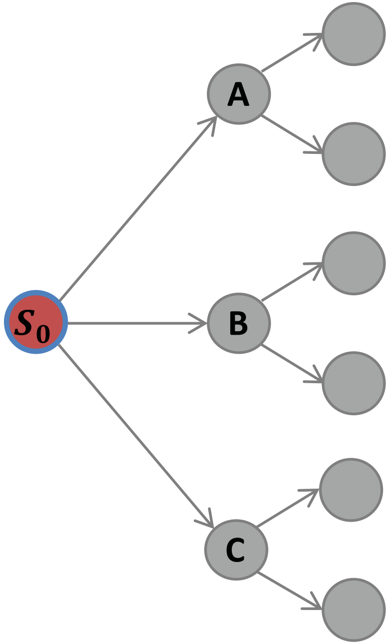
$$S_E = \{S_0\}$$

$$R_E(s,a) = \{S_0\text{:}20\}$$

$$T_E(s'|s,a) = \{S_0 \rightarrow \{A, B, C\}\text{:}0\}$$

$$\pi = \{S_0\text{:None}\}$$

Massachusetts Institute of Technology



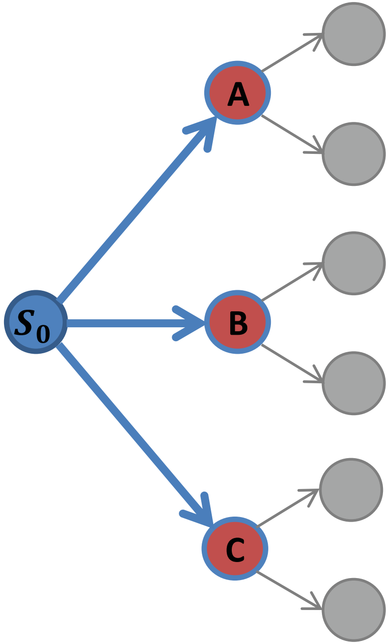
1. Create R_E and T_E
2. Find optimal policy on envelope
3. **Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$**
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

$$S_T = \{S_0\}$$

$$\pi = \{S_0\text{:None}\}$$

$$S_T \cap S^{\pi} = \{S_0\}$$

Massachusetts Institute of Technology



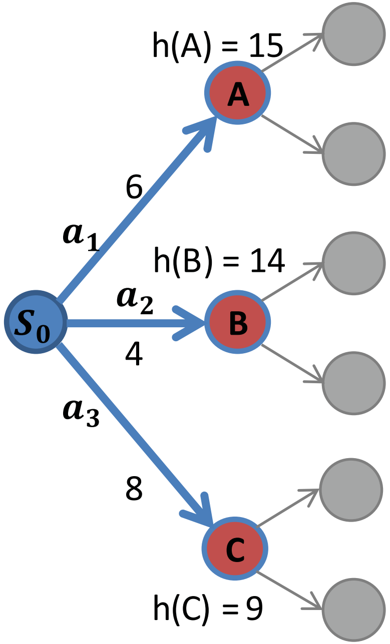
1. Create R_E and T_E
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. **Remove the expanded terminal states and add their children to the terminal state set**
5. **Add the new children to the envelope**
6. Repeat until no states are expanded

$$\boldsymbol{S_T} = \{\boldsymbol{S_T} \backslash \boldsymbol{S_0} \cup \text{children}(\boldsymbol{S_0}) \backslash \boldsymbol{S_E}\}$$

$$\boldsymbol{S_E} = \{\boldsymbol{S_E} \cup \text{children}(\boldsymbol{S_0})\}$$

$$\boldsymbol{S_T} = \{\boldsymbol{S_0}\} \to \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$$

$$\boldsymbol{S_E} = \{\boldsymbol{S_0}\} \to \{\boldsymbol{S_0}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$$



1. **Create R_E and T_E**
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

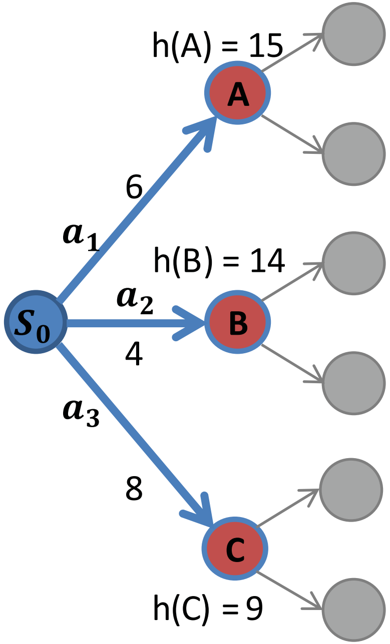
$$T_E(s'|s,a) = \begin{cases} 0 & \text{if } S \in S_T \\ \Pr(s'|s,a) & \text{otherwise} \end{cases}$$

$$R_E(s,a) = \begin{cases} h(s) & \text{if } S \in S_T \\ \mathrm{R}(s,a) & \text{otherwise} \end{cases}$$

$\boldsymbol{T_E(s'|s,a)} =$

$s = S_0$	s'=**A**	s'=**B**	s'=**C**
$\boldsymbol{a_1}$	*1*	*0*	*0*
$\boldsymbol{a_2}$	*0*	*1*	*0*
$\boldsymbol{a_3}$	*0*	*0*	*1*

$$\boldsymbol{R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8,}$$
$$\boldsymbol{(A,-) : 15, (B,-) : 14, (C,-) : 9\}}$$



1. **Create R_E and T_E**
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

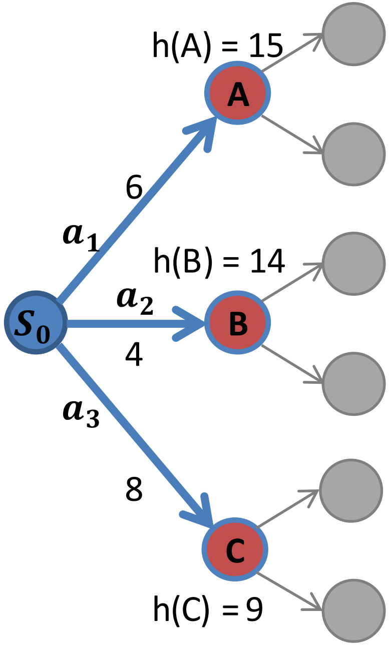
$$T_E(s'|s,a) = \begin{cases} 0 & \text{if } S \in S_T \\ \Pr(s'|s,a) & \text{otherwise} \end{cases}$$

$$R_E(s,a) = \begin{cases} h(s) & \text{if } S \in S_T \\ \mathrm{R}(s,a) & \text{otherwise} \end{cases}$$

$\boldsymbol{T_E(s'|s,a)} =$

$s = S_0$	s'=A	s'=B	s'=C
$\boldsymbol{a_1}$	***.98***	*.02*	*0*
$\boldsymbol{a_2}$	*.01*	***.99***	*0*
$\boldsymbol{a_3}$	*.05*	*.08*	***.87***

$$\boldsymbol{R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8,}$$
$$\boldsymbol{(A,-) : 15, (B,-) : 14, (C,-) : 9\}}$$



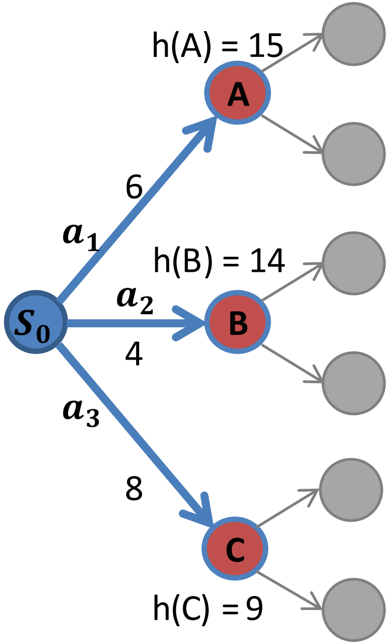
1. Create R_E and T_E
2. **Find optimal policy on envelope**
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

$$\pi\ ?$$

$T_E(s'|s,a) =$

$s = S_0$	s'=A	s'=B	s'=C
a_1	1	0	0
a_2	0	1	0
a_3	0	0	1

$$R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8,$$
$$(A,-) : 15, (B,-) : 14, (C,-) : 9\}$$



1. Create R_E and T_E
2. **Find optimal policy on envelope**
3. **Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$**
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

$$\pi = \{A\text{: None}, B\text{: None}, C\text{: None}, S_0 : a_1\}$$

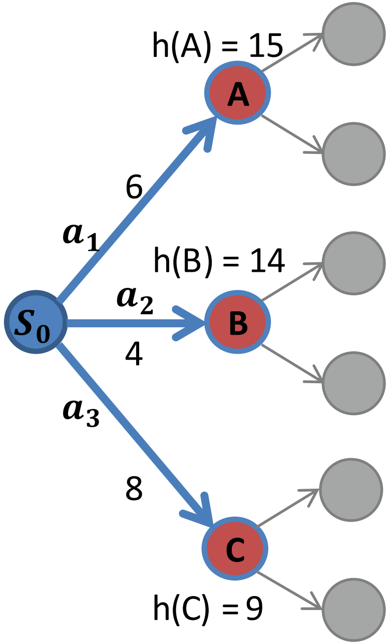
$$S_T{}^{\pi} = \{A\}$$

$$R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8,\ (A,-) : 15, (B,-) : 14, (C,-) : 9\}$$

$T_E(s'|s,a) =$

$s = S_0$	s'=**A**	s'=**B**	s'=**C**
a_1	*1*	*0*	*0*
a_2	*0*	*1*	*0*
a_3	*0*	*0*	*1*

Massachusetts Institute of Technology



1. Create R_E and T_E
2. **Find optimal policy on envelope**
3. **Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$**
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded

$$\pi = \{A\text{: None}, B\text{: None}, C\text{: None}, S_0 : a_1\}$$

$$S_T{}^{\pi} = \ ???$$

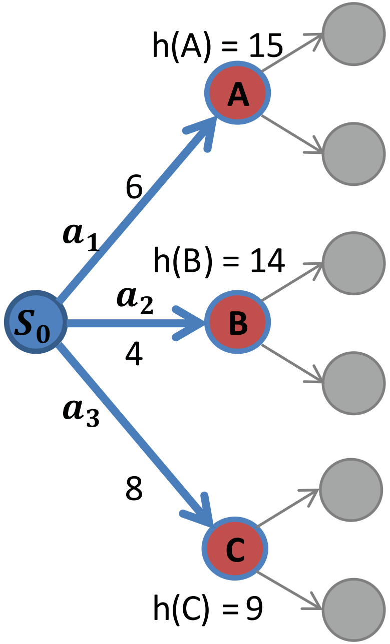
$T_E(s'|s,a) =$

$s = S_0$	s'=A	s'=B	s'=C
a_1	**.98**	.02	0
a_2	.01	**.99**	0
a_3	.05	.08	**.87**

$$R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8, (A,-) : 15, (B,-) : 14, (C,-) : 9\}$$

Massachusetts Institute of Technology

1. Create R_E and T_E
2. **Find optimal policy on envelope**
3. **Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$**
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. Repeat until no states are expanded



$$\pi = \{A\text{: None}, B\text{: None}, C\text{: None}, S_0 : a_1\}$$

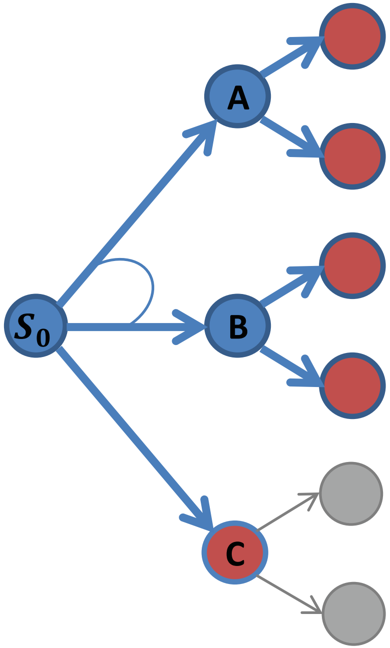
$$S_T{}^{\pi} = \{A, B\}$$

$T_E(s'|s,a) =$

$s = S_0$	s'=A	s'=B	s'=C
a_1	**.98**	.02	0
a_2	.01	**.99**	0
a_3	.05	.08	**.87**

$$R_E(s,a) = \{(S_0,a_1){:}\,6, (S_0,a_2){:}\,4, (S_0,a_3) : 8, (A,-) : 15, (B,-) : 14, (C,-) : 9\}$$

Massachusetts Institute of Technology

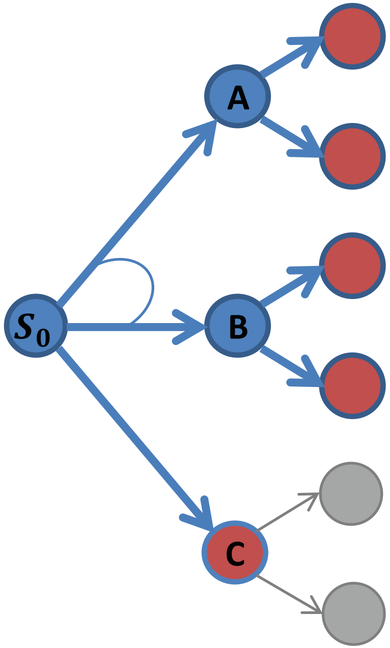


1. Create R_E and T_E
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. **Remove the expanded terminal states and add their children to the terminal state set**
5. **Add the new children to the envelope**
6. Repeat until no states are expanded

$T_E(s'|s,a) =$

$s = S_0$	s'=A	s'=B	s'=C
a_1	***.98***	*.02*	*0*
a_2	*.01*	***.99***	*0*
a_3	*.05*	*.08*	***.87***

Massachusetts Institute of Technology



1. Create R_E and T_E
2. Find optimal policy on envelope
3. Select the all the terminal states that are reachable by the optimal policy as $S_T{}^{\pi}$
4. Remove the expanded terminal states and add their children to the terminal state set
5. Add the new children to the envelope
6. **Repeat until no states are expanded**

$T_E(s'|s,a) =$

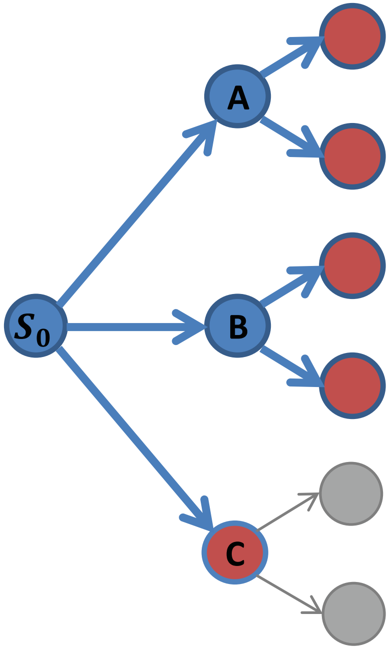
$s = S_0$	s'=**A**	s'=**B**	s'=**C**
a_1	***.98***	*.02*	*0*
a_2	*.01*	***.99***	*0*
a_3	*.05*	*.08*	***.87***

Massachusetts Institute of Technology

and add their children to the terminal state set

5. Add the new children to the envelope

6. **Repeat until no states are expanded**

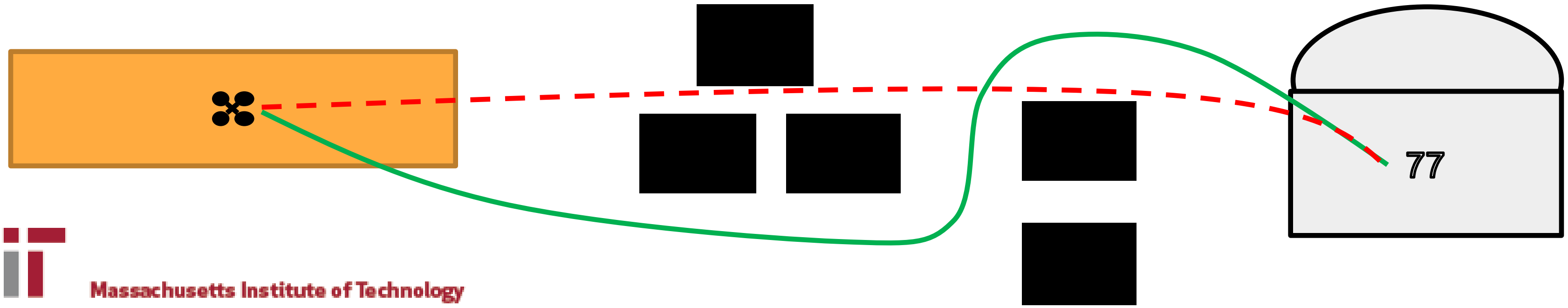


Termination:

- When the goal is reached and there are no more heuristically guided states to explore
- Envelope only grows to states that are “reachable” and “needed”
- Tighter probabilistic coupling, means more states need to be explored
    - If we don’t explore coupled states, we risk getting lost if we ever find ourselves there
- A complete policy for traversing the state space under uncertain transitions by chaining Value/Policy – Iteration and graph exploration

MDP Takeaways

1. Real platforms can be modelled stochastically
2. Real platforms sometimes must be modelled stochastically
3. Planning under uncertainty is possible but computationally expensive
4. A little bit of heuristic search helps a lot



Massachusetts Institute of Technology

Outline

1. Quadrotor motivating example
2. Planning with Markov Processes
    1. Markov Decision Process formulation
    2. Value Iteration Algorithm
    3. Heuristic-Guided solvers
3. Extensions to Partially Observable Markov Decision Processes
    1. Partially Observable Markov Decision Process formulation
    2. PRMs in the belief space
    3. Results from FIRM case study

Planning with Partially Observable Markov Decision Processes

Easier

Harder

Dynamics: deterministic
Sensors: deterministic

Dead reckoning, validate through sensing

Dynamics: deterministic
Sensors: stochastic

Dead reckoning, Kalman filtering

Dynamics: stochastic
Sensors: deterministic

Markov Decision Processes!

Dynamics: stochastic
Sensors: stochastic

Partially Observable Markov Decision Processes!

Massachusetts Institute of Technology

Motivating Example

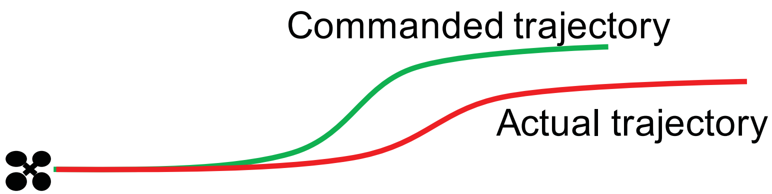
Quadrotor Motion Planning

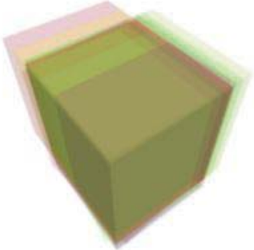
Action Uncertainty:

1) Noisy motors
2) Wind gusts
3) Dropped command signals

Observation Uncertainty:

1) Noisy camera
2) Changing lighting conditions
3) Obscured landmarks (d*-lite)
4) Non-unique environment





Noisy views of a cube

Partially Observable Markov Decision Processes (POMDPs)

A partially observable Markov decision process is defined as a tuple with seven elements:

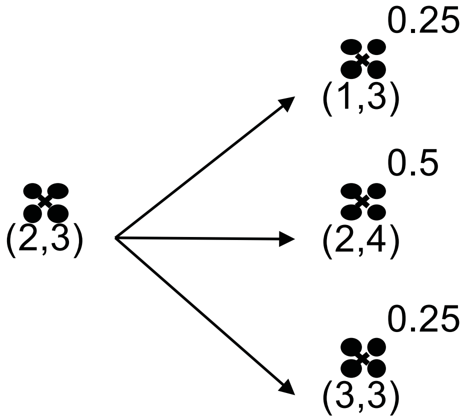
S: a set of states
A: a set of actions
T: a set of conditional transition probabilities between states
$R(s, a, s')$: an immediate reward for using action a in state s to go to state s'
Ω: a set of observations
O: a set of conditional observation probabilities
γ: a discount factor for rewards

maximize $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$

The robot is in state $s \epsilon S$, takes action $a \epsilon A$, which causes a tranisition to $s' \epsilon S$ with probability $T(s'|s, a)$. The robot then makes an observation $o \epsilon \Omega$, which depends on the new state with probability $O(o|s', a)$.

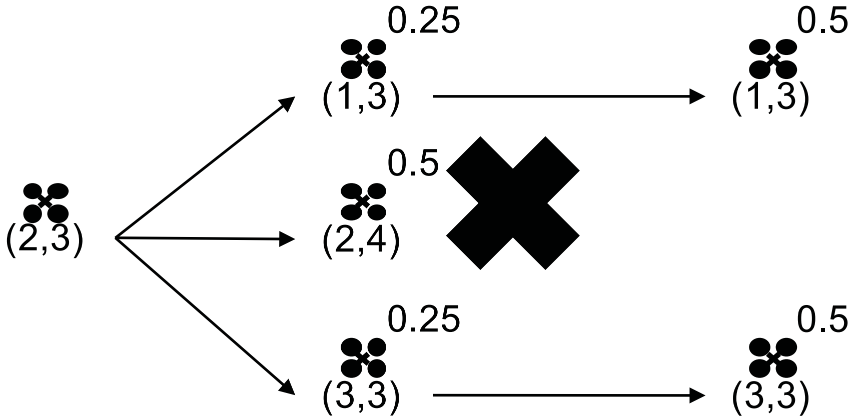
POMDP Example

$a_1 = N$



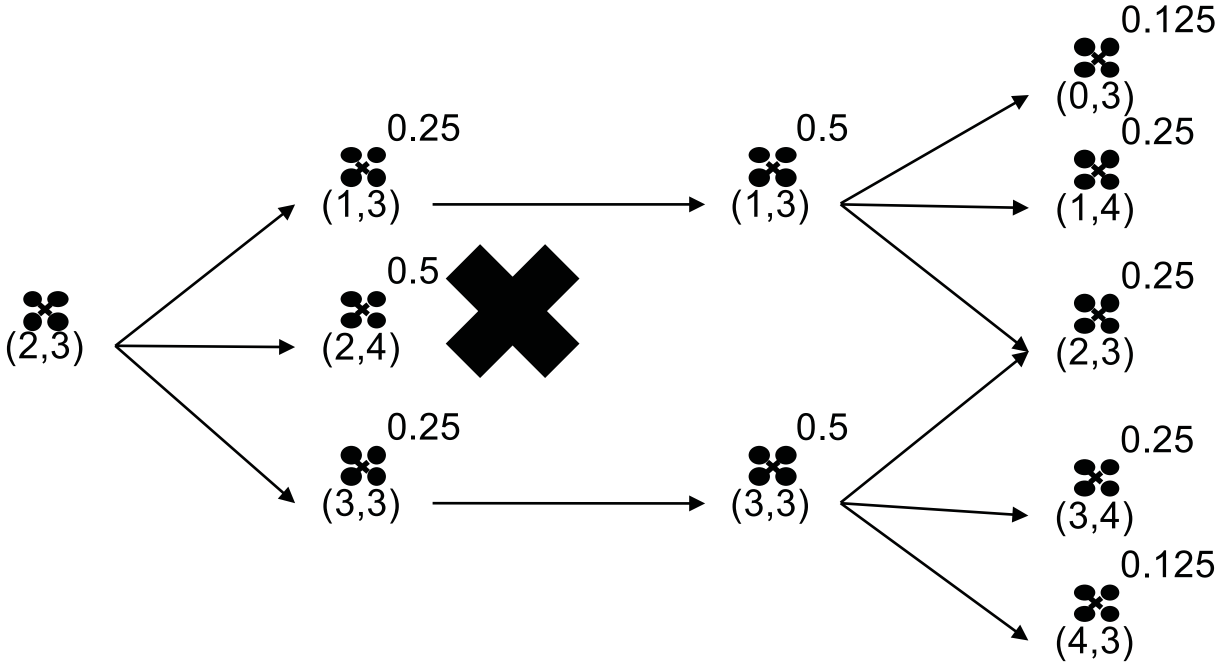
POMDP Example

$$a_1 = N, o_1 = (\cdot, 3)$$



POMDP Example

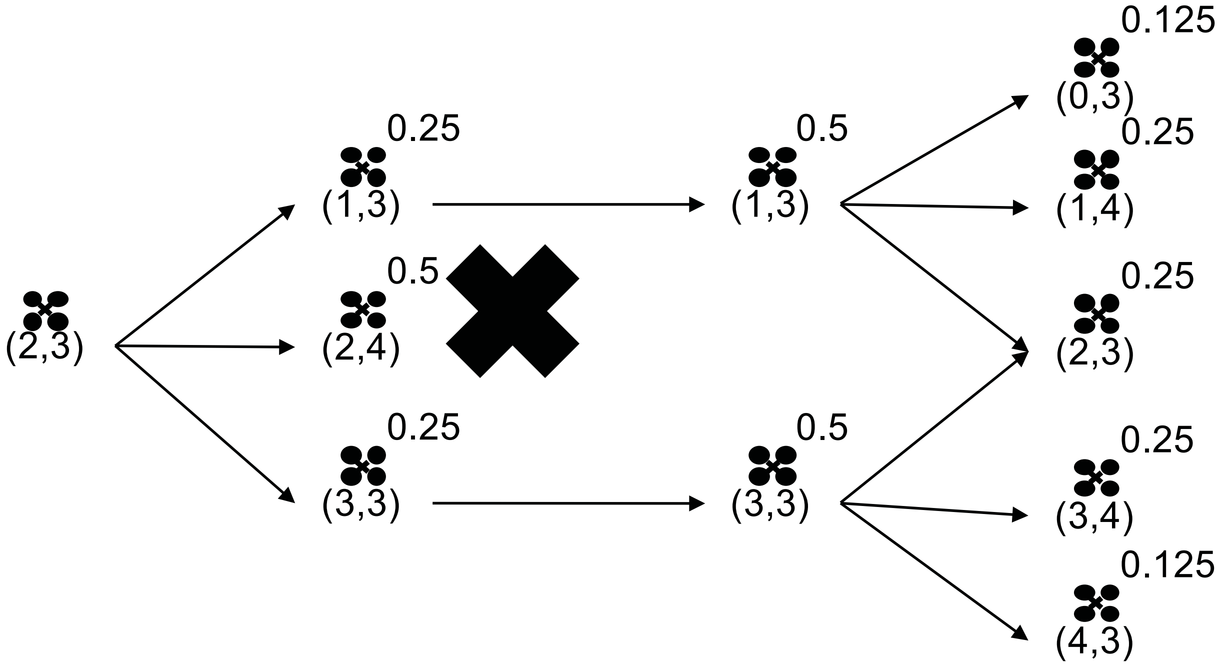
$a_1 = N, o_1 = (\cdot, 3), a_2 = N$



Massachusetts Institute of Technology

POMDP Example

$a_1 = N, o_1 = (\cdot, 3), a_2 = N$



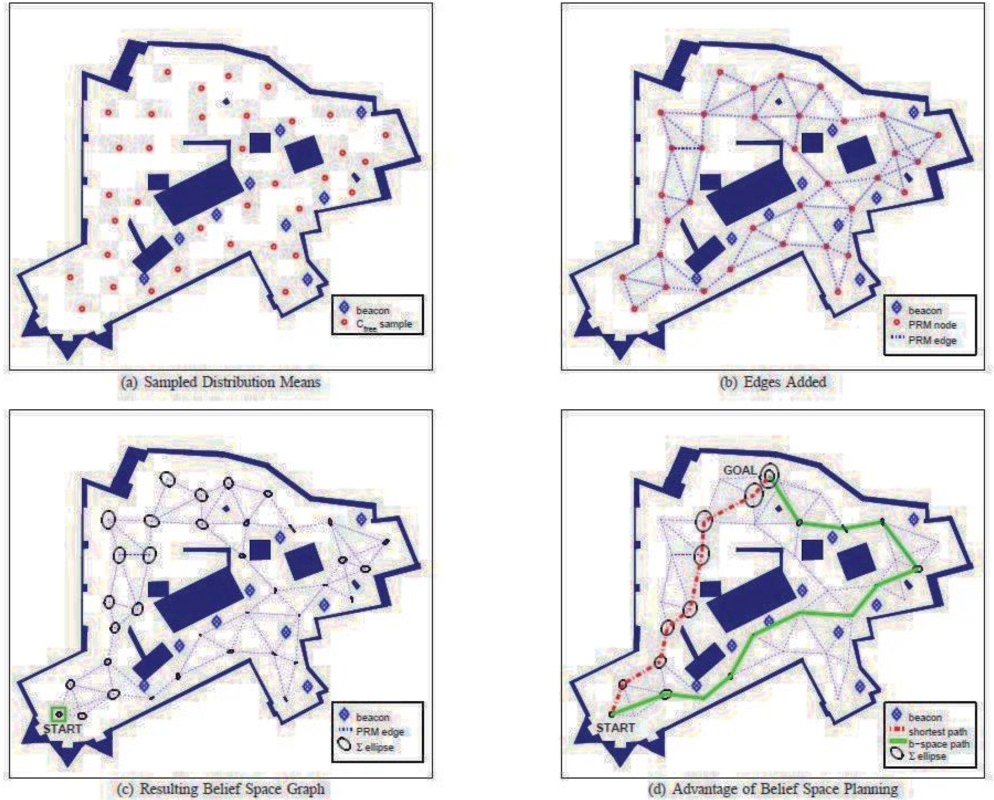
We cannot fully collapse our distribution even when we make observations

Planning in the Belief Space with PRMs

Goal is to generate a policy that maps from a belief state to an action

PRM

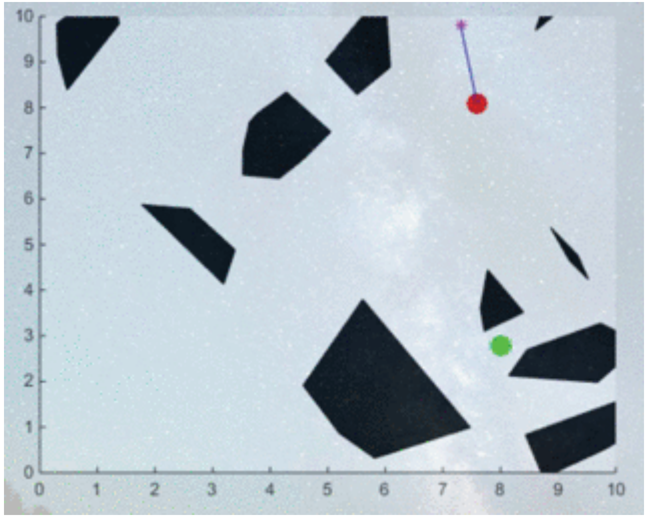
1) Sample points from c-space
2) Connect nearby points via collision-free edges
3) Transform c-space values to belief state values
4) Find shortest path on belief state graph



(a) Sampled Distribution Means (b) Edges Added (c) Resulting Belief Space Graph (d) Advantage of Belief Space Planning

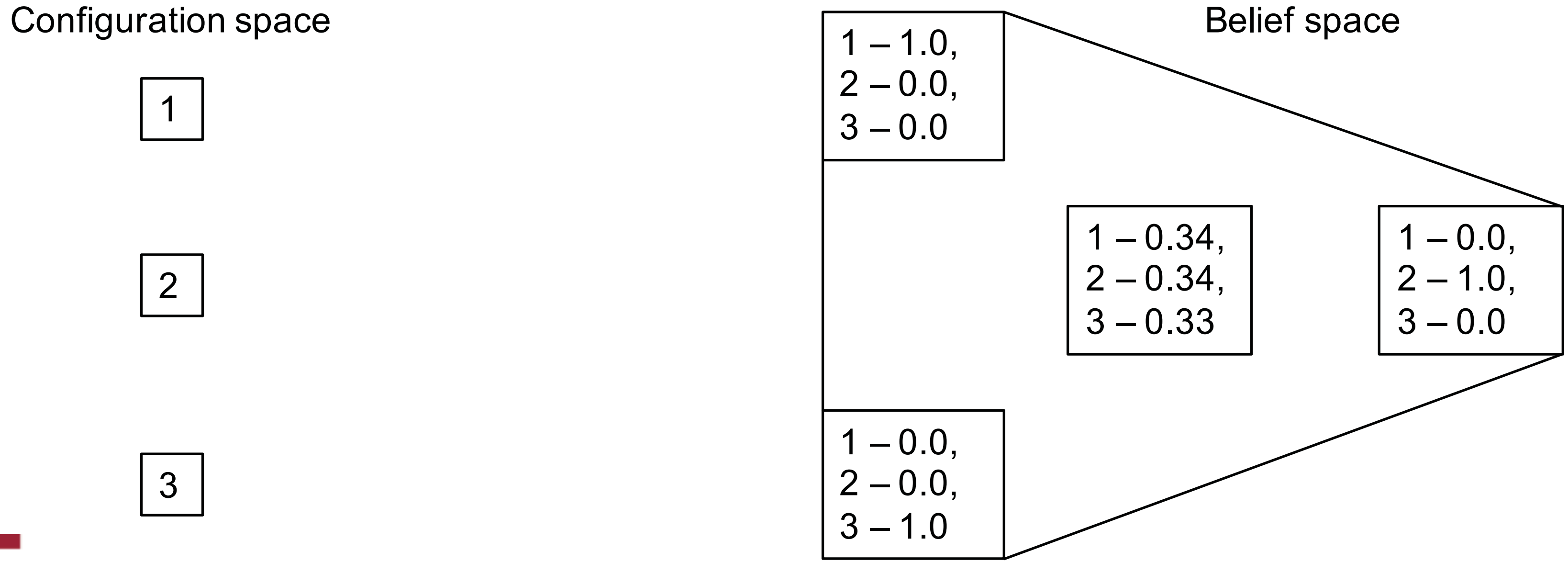
Configuration-Space PRMs

```
G = new Graph();
G.add(start);
G.add(goal)
while (num_nodes < MAX) {
    random_sample = sample_from_free();
    G.add(random_sample);
    G.connect_within_radius(random_sample, r);
}
return G;
//to find shortest path, search over G
```



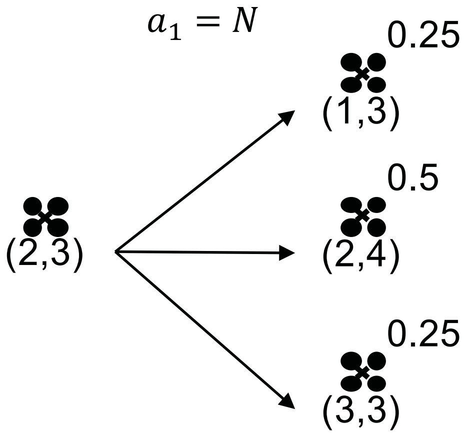
PRMs in the Belief Space

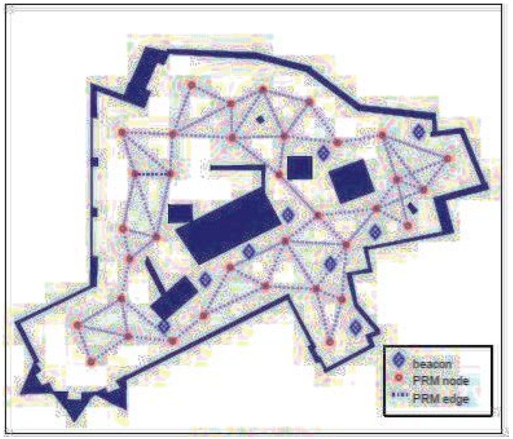
Do not have access to explicit configurations; instead have access to distributions over configurations (generally (μ, Σ))



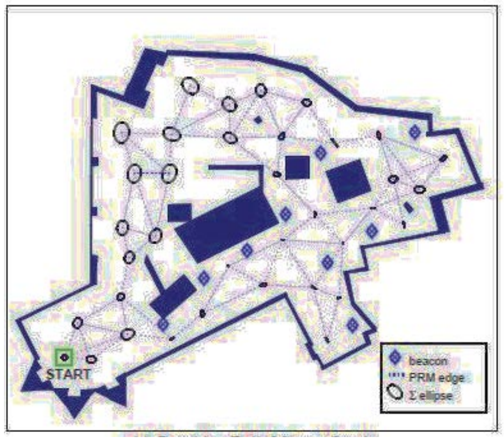
PRMs in the Belief Space

General strategy is to generate distributions from probabilistic models for each sampled node, but must make approximations for computational complexity





PRM



BRM

Case study: Feeback-based Information-state Roadmaps (FIRM)

1) Sample configurations μ

2) Build LQR controller around point → generate Σ

3) Connect LQR regions via feedback controllers

$$Cost(B) = \alpha * E[time] + \beta * Uncertainty$$

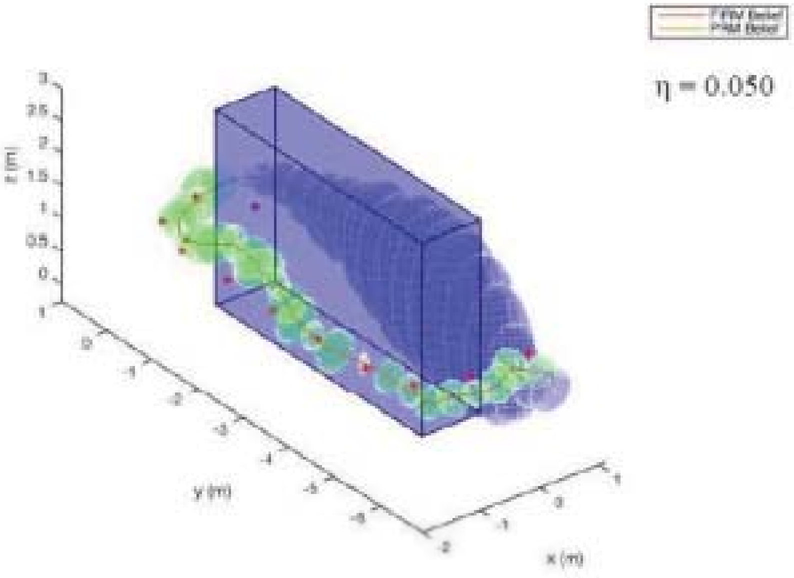
Massachusetts Institute of Technology

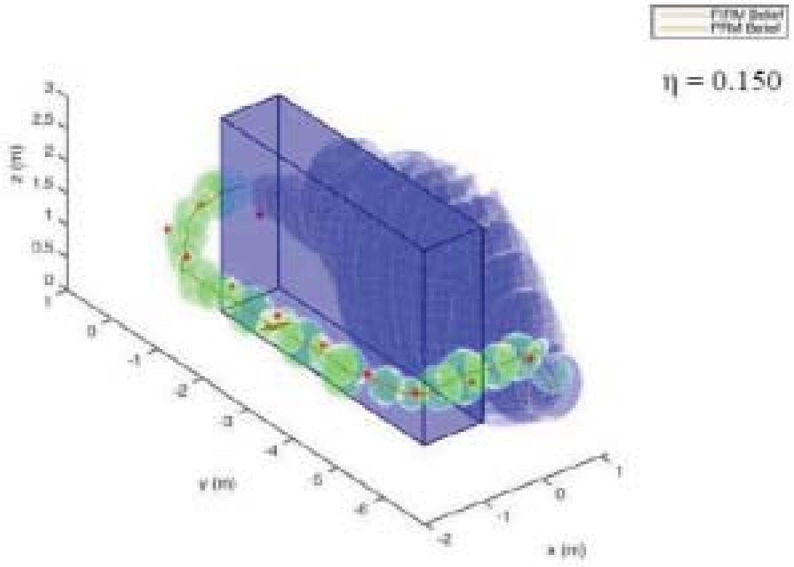
Case study: FIRM

$$\underbrace{J(B_i)}_{\text{cost-to-go from belief node } B_i \text{ to the destination}} = \min_j \overbrace{C^{\mu_j}(B_i)}^{\text{cost of using controller } \mu_j \text{ from } B_i} + \underbrace{J(F)}_{\text{cost of colliding with an obstacle}} \overbrace{P^{\mu_j}(F|B_i)}^{\text{probability of colliding with an obstacle}} + \sum_m J(B_m) P^{\mu_j}(B_m, \bar{F} | B_i)$$

Case study: FIRM

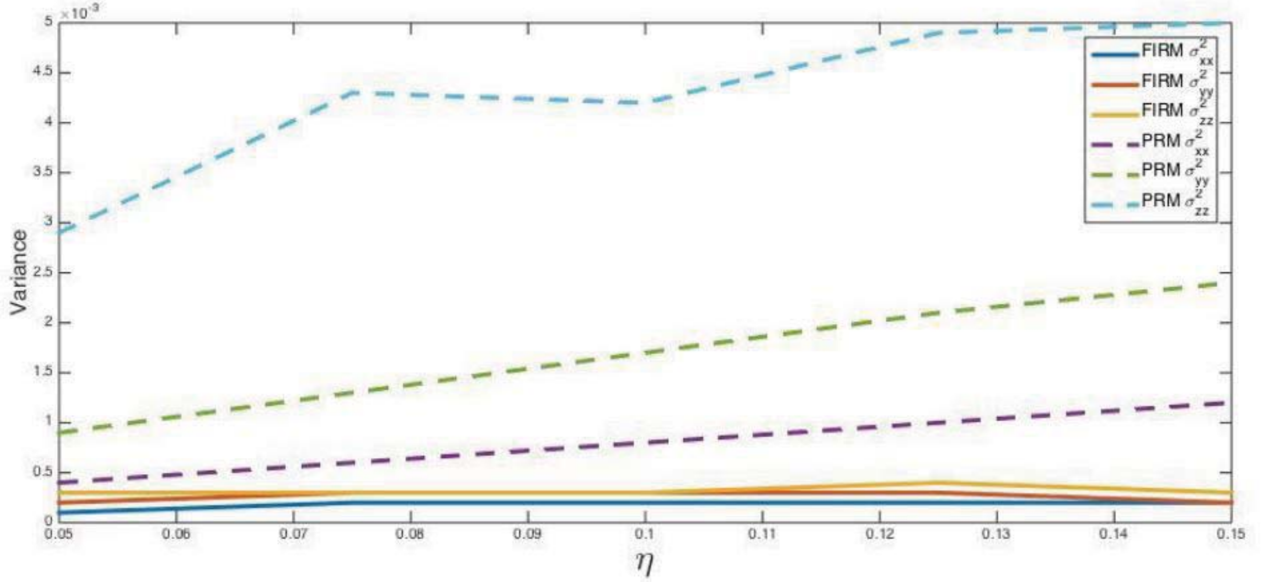
Results: FIRM prefers safer paths





Case study: FIRM

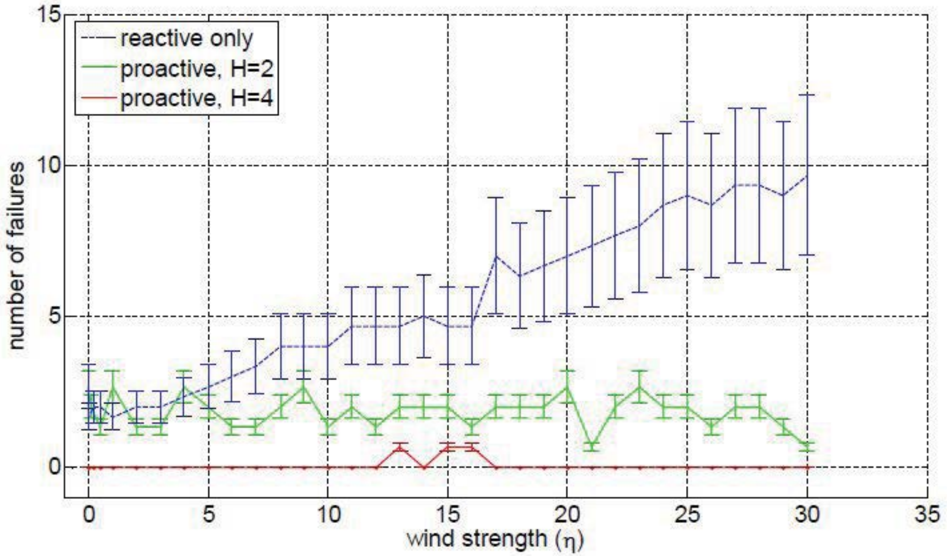
Results: FIRM minimizes state uncertainty as environmental noise grows



Massachusetts Institute of Technology

Case study: FIRM

Results: FIRM maintains lower uncertainty, improving performance overall

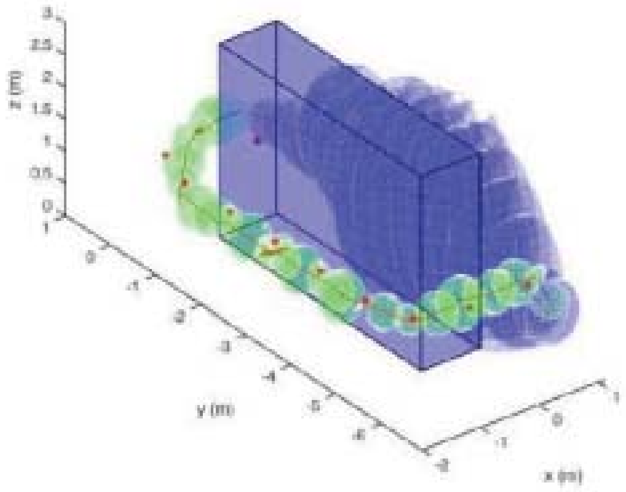


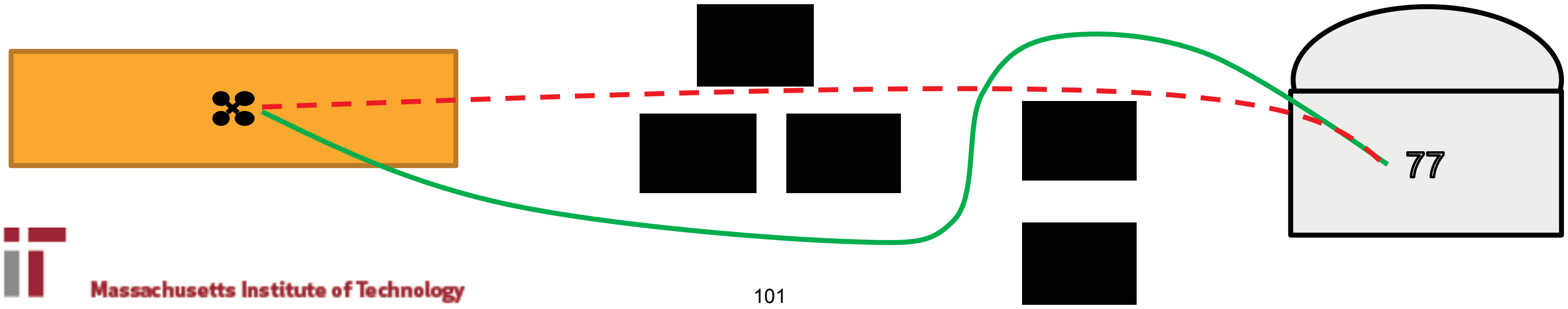
Probabilistic Takeaways

1) Real world processes are stochastic

2) Solving stochastic problems is far more complex

3) A little thought (heuristics, assumptions) helps a lot





Resources

Russ Tedrake. *Underactuated Robotics: Algorithmsfor Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832).* Downloadedon04/26/2016from http://underactuated.mit.edu/

B.C. Williams, *Lecture 8a Markov Decision Processes: Reactive Planning to Maximize Reward*, https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2016/lecture-notes/16-412s16ResourceFile.pdf

Eric A. Hansen, Shlomo Zilberstein, LAO∗: A heuristic search algorithm that finds solutions with loops, Artificial Intelligence, Volume 129, Issues 1–2, June 2001, Pages 35-62, ISSN 0004-3702, http://dx.doi.org/10.1016/S0004-3702(01)00106-0.

He R., Prentice S., Roy N., Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment, ICRA, 2008, Pages 1814-1820, ISSN 1050-4729, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4543471.

Agha-mohammadi A.,Ure N., How J., Vian J,Health Aware Stochastic Planning For Persistent Package Delivery Missions using Quadrotors, IEEE, 2014, Pages 3389-3396,ISSN2153-0858, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6943034.

Pascal Poupart, Sequential Decision Making and Reinforcement Learning, 2013, University of Waterloo, https://cs.uwaterloo.ca/~ppoupart/teaching/cs886-spring13/slides/cs886-module9-lao-star.pdf.

Andrey Kolobov,Mausam, Probabilistic Planning with Markov Decision Processes, University of Washington, http://research.microsoft.com/en-us/um/people/akolobov/MDPs_Tutorial.pdf.