CS 747, Autumn 2022: Lecture 11

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2022

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

3. Review of MDP planning

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

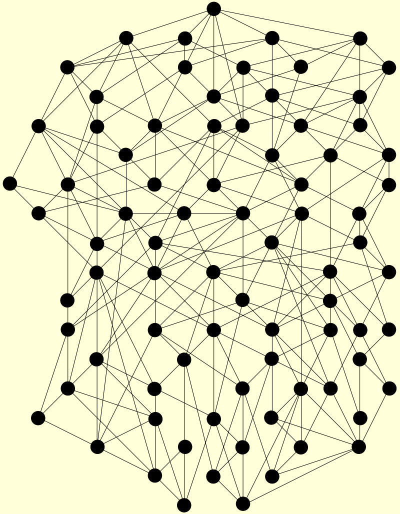
3. Review of MDP planning

Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.

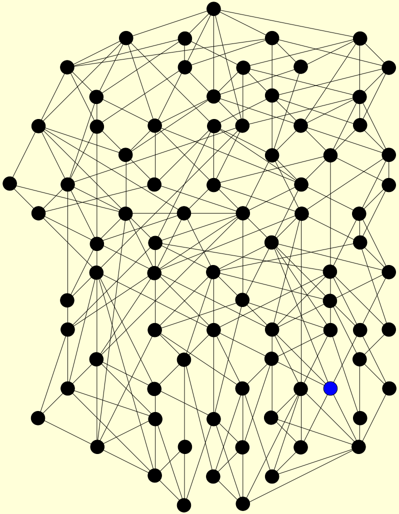
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π).
    $\pi \leftarrow \pi'$.
Return π.



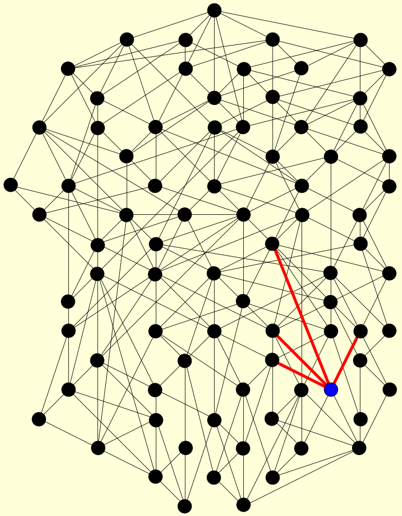
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.



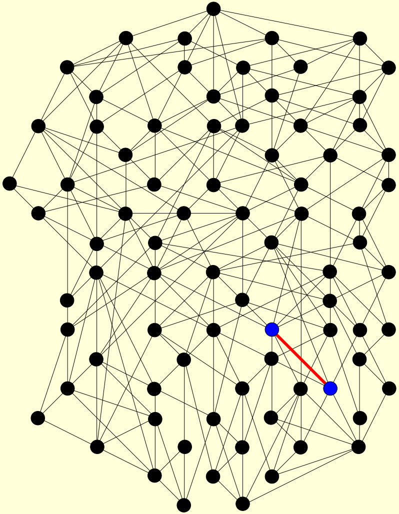
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.



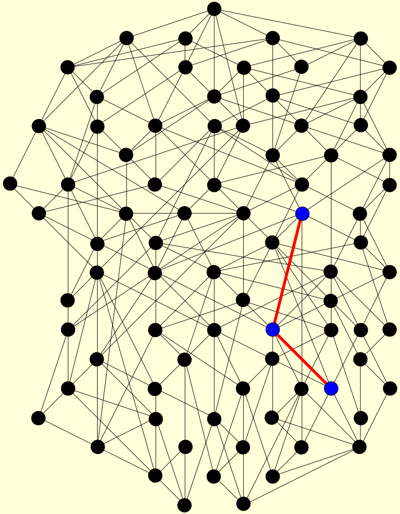
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.



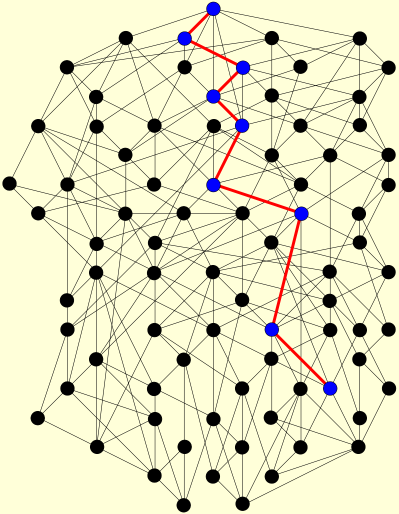
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.



Policy Iteration Algorithm

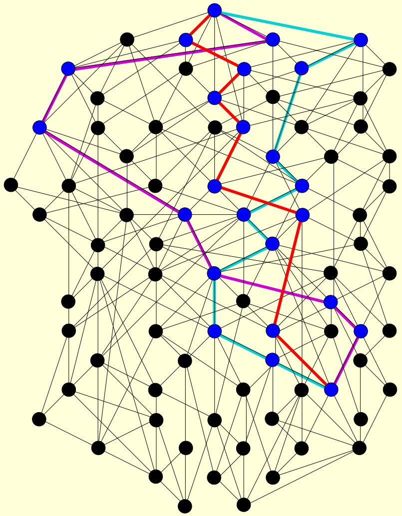
$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π).
  $\pi \leftarrow \pi'$.
Return π.



Policy Iteration Algorithm

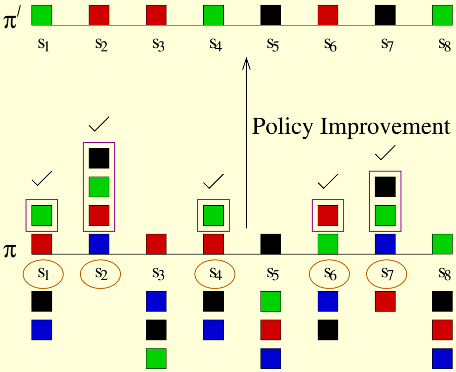
$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π).
    $\pi \leftarrow \pi'$.
Return π.

Path taken (and hence the number of iterations) in general depends on the switching strategy.



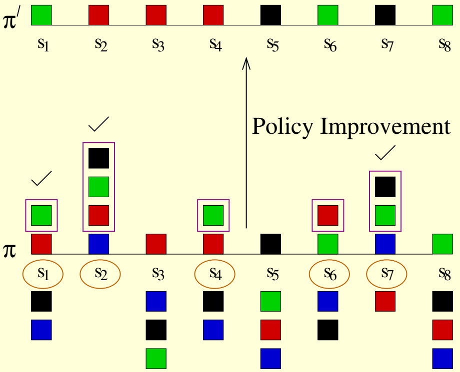
Howard's Policy Iteration

- Reference: Howard (1960).
- Greedy; switch all improvable states.



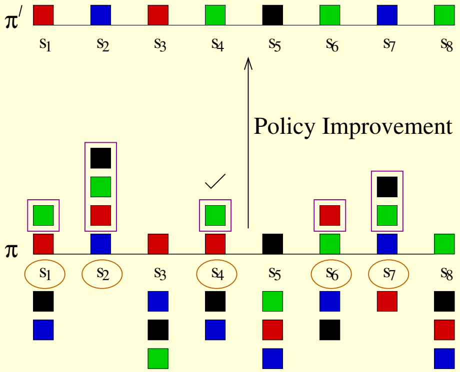
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



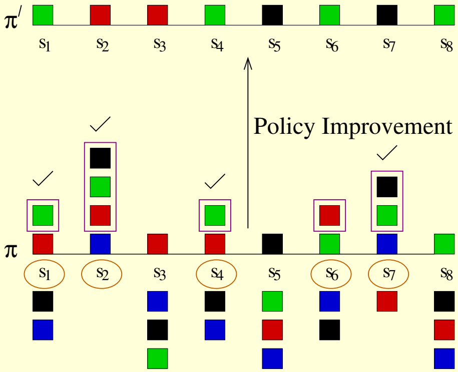
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



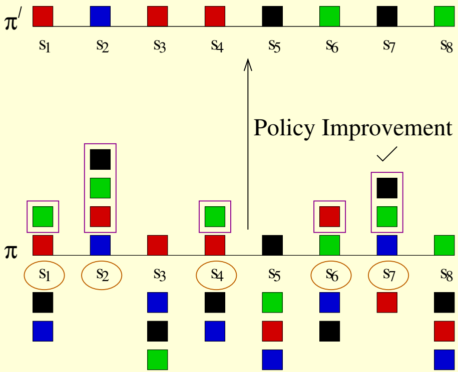
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



Simple Policy Iteration

- Reference: Melekopoglou and Condon (1994).
- Assume a fixed indexing of states.
- Switch the improvable state with the highest index.



Upper and Lower Bounds

$U(n, k)$ is an upper bound applicable to a set of PI variants $\mathcal{L}$ if

- for each n-state, k-action MDP $M = (S, A, T, R, \gamma)$,
- for each policy $\pi : S \to A$,
- for each algorithm $L \in \mathcal{L}$,

the expected number of policy evaluations performed by L on M if initialised at π is at most $U(n, k)$.

Upper and Lower Bounds

$U(n, k)$ is an upper bound applicable to a set of PI variants $\mathcal{L}$ if

- for each n-state, k-action MDP $M = (S, A, T, R, \gamma)$,
- for each policy $\pi : S \rightarrow A$,
- for each algorithm $L \in \mathcal{L}$,

the expected number of policy evaluations performed by L on M if initialised at π is at most $U(n, k)$.

$L(n, k)$ is a lower bound applicable to a set of PI variants $\mathcal{L}$ if

- there exists an n-state, k-action MDP $M = (S, A, T, R, \gamma)$,
- there exists a policy $\pi : S \rightarrow A$,
- there exists an algorithm $L \in \mathcal{L}$,

such that the expected number of policy evaluations performed by L on M if initialised at π is at least $L(n, k)$.

Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$
Mansour and Singh's Random PI [HPZ14]	Randomised	$\text{poly}(n) \cdot 1.5^n$	–

Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$
Mansour and Singh's Random PI [HPZ14]	Randomised	$\text{poly}(n) \cdot 1.5^n$	–

Lower bounds on number of iterations

$\Omega(n)$ Howard's PI on n-state, 2-action MDPs [HZ10].

Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$
Mansour and Singh's Random PI [HPZ14]	Randomised	$\text{poly}(n) \cdot 1.5^n$	–

Lower bounds on number of iterations

$\Omega(n)$ Howard's PI on n-state, 2-action MDPs [HZ10].

$\Omega(2^n)$ Simple PI on n-state, 2-action MDPs [MC94].

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n for $k = 2$.

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n for $k = 2$.
- Gupta and Kalyanakrishnan (2017) give a deterministic PI variant with upper bound $k^{0.7207n}$. Taraviya and Kalyanakrishnan (2019) improve to $k^{0.7019n}$.

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n for $k = 2$.
- Gupta and Kalyanakrishnan (2017) give a deterministic PI variant with upper bound $k^{0.7207n}$. Taraviya and Kalyanakrishnan (2019) improve to $k^{0.7019n}$.

- Kalyanakrishnan, Misra, and Gopalan (2016) show an upper bound of $(2 + \ln(k-1))^n$ for a randomised PI variant.

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n for $k = 2$.
- Gupta and Kalyanakrishnan (2017) give a deterministic PI variant with upper bound $k^{0.7207n}$. Taraviya and Kalyanakrishnan (2019) improve to $k^{0.7019n}$.

- Kalyanakrishnan, Misra, and Gopalan (2016) show an upper bound of $(2 + \ln(k - 1))^n$ for a randomised PI variant.
- Taraviya and Kalyanakrishnan (2019) show an upper bound of $(O(\sqrt{k \log(k)}))^n$ for a randomised variant of Howard's PI.

PI: Some Recent Results

(Polynomial factors ignored. Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n for $k = 2$.
- Gupta and Kalyanakrishnan (2017) give a deterministic PI variant with upper bound $k^{0.7207n}$. Taraviya and Kalyanakrishnan (2019) improve to $k^{0.7019n}$.

- Kalyanakrishnan, Misra, and Gopalan (2016) show an upper bound of $(2 + \ln(k-1))^n$ for a randomised PI variant.
- Taraviya and Kalyanakrishnan (2019) show an upper bound of $(O(\sqrt{k \log(k)}))^n$ for a randomised variant of Howard's PI.

- Ashutosh, Consul, Dedhia, Khirwadkar, Shah, and Kalyanakrishnan (2020) show a *lower bound* of $\sqrt{k}^n$ iterations for a deterministic variant of PI.

Markov Decision Problems

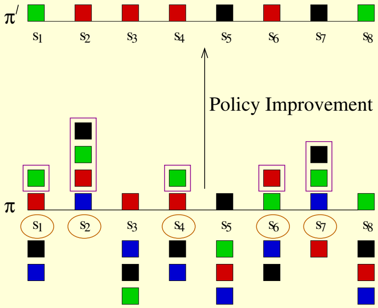
1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
    - Basic tools
    - Howard's PI with $k = 2$
    - BSPI with $k = 2$
    - Open problems

3. Review of MDP planning

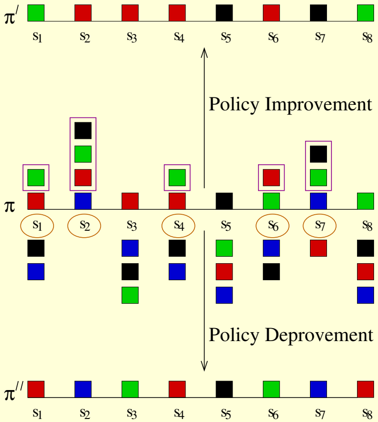
1. Policy Improvement and Policy "Deprovement"

$$\boxed{\pi' \succ \pi.}$$



1. Policy Improvement and Policy "Deprovement"

$$\pi' \succ \pi.$$



$$\pi \succeq \pi''.$$

2. Improvement sets in 2-action MDPs

Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

2. Improvement sets in 2-action MDPs

Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

$\boxed{1}$ $\boxed{0}$ $\boxed{1}$ $\boxed{1}$ 0 1 1 0 $\boxed{1}$ $\boxed{1}$ $\boxed{0}$ $\boxed{0}$ 1 1 0 1

2. Improvement sets in 2-action MDPs

Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

1 1 0 0 0 1 1 0

$\succ$

$\boxed{\mathbf{1}}$ $\boxed{\mathbf{0}}$ $\boxed{\mathbf{1}}$ $\boxed{\mathbf{1}}$ **0 1 1 0** $\boxed{\mathbf{1}}$ $\boxed{\mathbf{1}}$ $\boxed{\mathbf{0}}$ $\boxed{\mathbf{0}}$ **1 1 0 1**

2. Improvement sets in 2-action MDPs

Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

1 1 0 0 0 1 1 0

$\succ$

$\boxed{1}$ $\boxed{0}$ $\boxed{1}$ $\boxed{1}$ **0 1 1 0** $\boxed{1}$ $\boxed{1}$ $\boxed{0}$ $\boxed{0}$ **1 1 0 1**

$\succeq$

1 0 1 1 1 1 0 1

2. Improvement sets in 2-action MDPs

Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

1 1 0 0 0 1 1 0	1 0 1 1 1 1 0 1
$\succ$	$\succ$
$\boxed{1}$ $\boxed{0}$ $\boxed{1}$ $\boxed{1}$ 0 1 1 0	$\boxed{1}$ $\boxed{1}$ $\boxed{0}$ $\boxed{0}$ 1 1 0 1
$\succeq$	
1 0 1 1 1 1 0 1	

2. Improvement sets in 2-action MDPs

> Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

$$\begin{array}{cccccccccccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \quad & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
 & & & \succ & & & & & & & & & \succ & & & & \\
\boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} & 0 & 1 & 1 & 0 & \quad & \boxed{1} & \boxed{1} & \boxed{0} & \boxed{0} & 1 & 1 & 0 & 1 \\
 & & & \succeq & & & & & & & & & \succeq & & & & \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \quad & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}$$

2. Improvement sets in 2-action MDPs

> Non-optimal policies $\pi, \pi' \in \Pi$ cannot have the same set of improvable states.

$$\begin{array}{cccccccccccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
 & & & \succ & & & & & & & & & \succ & & & & \\
\boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} & 0 & 1 & 1 & 0 & & \boxed{1} & \boxed{1} & \boxed{0} & \boxed{0} & 1 & 1 & 0 & 1 \\
 & & & \succeq & & & & & & & & & \succeq & & & & \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}$$

Contradiction!

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

3. Review of MDP planning

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π 0 0 0 0 0 0 0 $\boxed{0}$ $\boxed{0}$ $\boxed{0}$ $\boxed{0}$ $\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
π	**0**	**0**	**0**	**0**	**0**	**0**	**0**	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

Possible?

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	1
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	1
π_1	0	0	0	0	0	0	0	1	1	1	1	0
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	1
π_1	0	0	0	0	0	0	0	1	1	1	1	0
π_2	0	0	0	0	0	0	0	1	1	1	0	0
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	1
π_1	0	0	0	0	0	0	0	1	1	1	1	0
π_2	0	0	0	0	0	0	0	1	1	1	0	0
π_3	0	0	0	0	0	0	0	1	1	0	0	0
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

$$
\begin{array}{lcccccccccccc}
\pi' & 0 & \boxed{0} & 0 & 0 & \boxed{0} & 0 & 0 & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & 1 \\
\pi_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\pi_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
\pi_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
\pi_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\pi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0}
\end{array}
$$

Howard's Policy Iteration (2-action MDPs)

Switch actions in every improvable state.

π'	0	$\boxed{0}$	0	0	$\boxed{0}$	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	1
π_1	0	0	0	0	0	0	0	1	1	1	1	0
π_2	0	0	0	0	0	0	0	1	1	1	0	0
π_3	0	0	0	0	0	0	0	1	1	0	0	0
π_4	0	0	0	0	0	0	0	1	0	0	0	0
π	0	0	0	0	0	0	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

If π has *m* improvable states and $\pi \xrightarrow{\text{Howard's PI}} \pi'$, then there exist *m* policies π'' such that $\pi' \succeq \pi'' \succ \pi$.

Howard's Policy Iteration (2-action MDPs)

- Take $m^{\star} = \frac{n}{3}$.

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

$$\leq \frac{2^n}{m^\star} = \frac{2^n}{n/3}.$$

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

$$\leq \frac{2^n}{m^\star} = \frac{2^n}{n/3}.$$

- Number of policies with fewer than $m^\star$ improvable states visited

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

$$\leq \frac{2^n}{m^\star} = \frac{2^n}{n/3}.$$

- Number of policies with fewer than $m^\star$ improvable states visited

$$\leq \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{m^\star - 1}$$

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

$$\leq \frac{2^n}{m^\star} = \frac{2^n}{n/3}.$$

- Number of policies with fewer than $m^\star$ improvable states visited

$$\leq \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{m^\star - 1} \leq 3\frac{2^n}{n}.$$

Howard's Policy Iteration (2-action MDPs)

- Take $m^\star = \frac{n}{3}$.
- Number of policies with $m^\star$ or more improvable states visited

$$\leq \frac{2^n}{m^\star} = \frac{2^n}{n/3}.$$

- Number of policies with fewer than $m^\star$ improvable states visited

$$\leq \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{m^\star - 1} \leq 3\frac{2^n}{n}.$$

Number of iterations taken by Howard's PI: $O\left(\frac{2^n}{n}\right)$ [MS99, HGDJ14].

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

3. Review of MDP planning

Batch-Switching Policy Iteration (BSPI) (2-action MDPs)

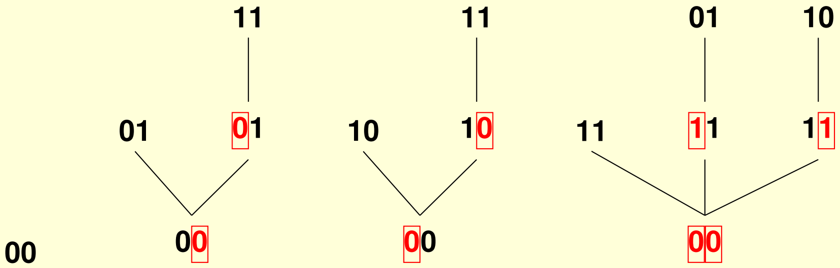
Howard's Policy Iteration takes at most ____ iterations on a 2-state MDP!

Batch-Switching Policy Iteration (BSPI) (2-action MDPs)

Howard's Policy Iteration takes at most _3_ iterations on a 2-state MDP!

Batch-Switching Policy Iteration (BSPI) (2-action MDPs)

Howard's Policy Iteration takes at most _3_ iterations on a 2-state MDP!



Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.

Given a policy, improve the rightmost set containing an improvable state.

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.

Given a policy, improve the rightmost set containing an improvable state.

π_1	**0**	$\boxed{\mathbf{1}}$	**1**	**0**	$\boxed{\mathbf{0}}$	$\boxed{\mathbf{0}}$	**1**	$\boxed{\mathbf{0}}$	$\boxed{\mathbf{0}}$	**0**
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.
Given a policy, improve the rightmost set containing an improvable state.

π_2	**0**	$\boxed{1}$	**1**	**0**	$\boxed{0}$	$\boxed{0}$	**1**	**0**	**1**	**0**
π_1	**0**	$\boxed{1}$	**1**	**0**	$\boxed{0}$	$\boxed{0}$	**1**	$\boxed{0}$	$\boxed{0}$	**0**
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.

Given a policy, improve the rightmost set containing an improvable state.

π_3	$\boxed{0}$	$\boxed{1}$	1	0	1	$\boxed{1}$	1	$\boxed{0}$	1	0
π_2	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	0	1	0
π_1	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	$\boxed{0}$	$\boxed{0}$	0
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.

Given a policy, improve the rightmost set containing an improvable state.

π_4	$\boxed{0}$	1	1	0	1	1	1	1	1	0
π_3	$\boxed{0}$	$\boxed{1}$	1	0	1	$\boxed{1}$	1	$\boxed{0}$	1	0
π_2	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	0	1	0
π_1	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	$\boxed{0}$	$\boxed{0}$	0
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.
Given a policy, improve the rightmost set containing an improvable state.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
π_4	$\boxed{0}$	1	1	0	1	1	1	1	1	0
π_3	$\boxed{0}$	$\boxed{1}$	1	0	1	$\boxed{1}$	1	$\boxed{0}$	1	0
π_2	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	0	1	0
π_1	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	$\boxed{0}$	$\boxed{0}$	0

- Left-most batch can change only when all other columns are non-improvable.

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.
Given a policy, improve the rightmost set containing an improvable state.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
π_4	$\boxed{0}$	1	1	0	1	1	1	1	1	0
π_3	$\boxed{0}$	$\boxed{1}$	1	0	1	$\boxed{1}$	1	$\boxed{0}$	1	0
π_2	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	0	1	0
π_1	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	$\boxed{0}$	$\boxed{0}$	0

- Left-most batch can change only when all other columns are non-improvable.
- Left-most batch can change at most 3 times (following previous result).

Batch-Switching Policy Iteration (BSPI)

Partition the states into 2-sized batches; arranged from left to right.
Given a policy, improve the rightmost set containing an improvable state.

π_4	$\boxed{0}$	1	1	0	1	1	1	1	1	0
π_3	$\boxed{0}$	$\boxed{1}$	1	0	1	$\boxed{1}$	1	$\boxed{0}$	1	0
π_2	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	0	1	0
π_1	0	$\boxed{1}$	1	0	$\boxed{0}$	$\boxed{0}$	1	$\boxed{0}$	$\boxed{0}$	0
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}

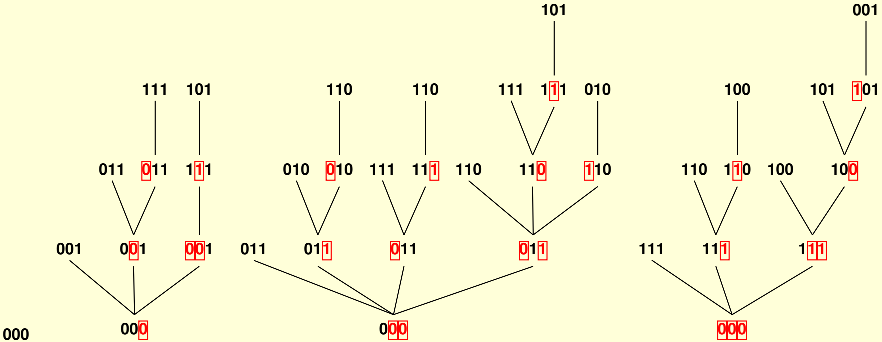
- Left-most batch can change only when all other columns are non-improvable.
- Left-most batch can change at most 3 times (following previous result).
- $T(n) \leq 3 \times T(n-2) \leq \sqrt{3}^{n}$.

Batch-Switching Policy Iteration (BSPI)

Howard's Policy Iteration takes at most 5 iterations on a 3-state MDP!

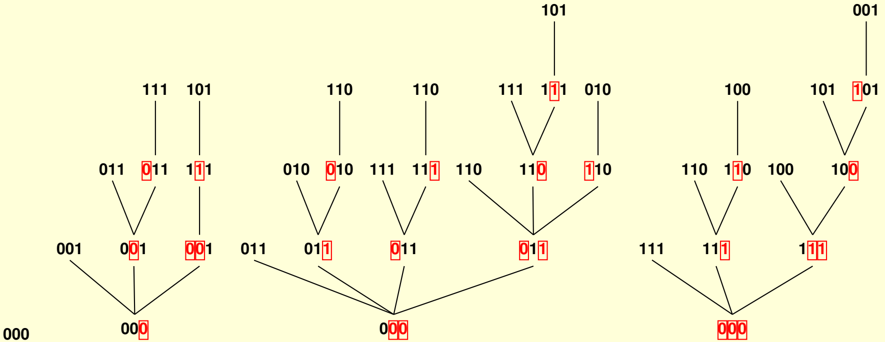
Batch-Switching Policy Iteration (BSPI)

Howard's Policy Iteration takes at most 5 iterations on a 3-state MDP!



Batch-Switching Policy Iteration (BSPI)

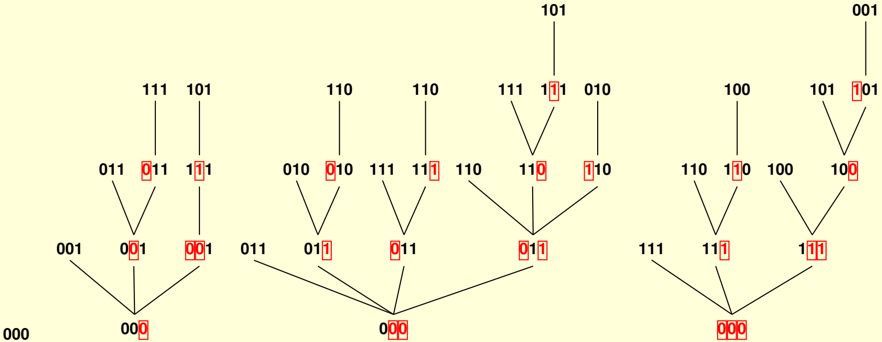
Howard's Policy Iteration takes at most 5 iterations on a 3-state MDP!



The structures above are called Trajectory-bounding Trees (TBTs) [KMG16a] (and correspond to the Order Regularity Problem [H12, GHDJ15]).

Batch-Switching Policy Iteration (BSPI)

Howard's Policy Iteration takes at most 5 iterations on a 3-state MDP!



The structures above are called Trajectory-bounding Trees (TBTs) [KMG16a] (and correspond to the Order Regularity Problem [H12, GHDJ15]).

BSPI with 3-sized batches gives $T(n) \leq 5 \times T(n-3) \leq 1.71^n$.

Upper Bounds

Batch size	Depth of TBT	Bound on number of iterations
1	2	2^n
2	3	1.7321^n
3	5	1.7100^n
4	8	1.6818^n
5	13	1.6703^n
6	21	1.6611^n
7	33	1.6479^n

Upper Bounds

Batch size	Depth of TBT	Bound on number of iterations
1	2	2^n
2	3	1.7321^n
3	5	1.7100^n
4	8	1.6818^n
5	13	1.6703^n
6	21	1.6611^n
7	33	1.6479^n

Depth of TBT for batch size 7 due to Gerencsér *et al.* [GHDJ15].

Upper Bounds

Batch size	Depth of TBT	Bound on number of iterations
1	2	2^n
2	3	1.7321^n
3	5	1.7100^n
4	8	1.6818^n
5	13	1.6703^n
6	21	1.6611^n
7	33	1.6479^n

Depth of TBT for batch size 7 due to Gerencsér *et al.* [GHDJ15].

Will the bound continue to be non-increasing in the batch size?

Upper Bounds

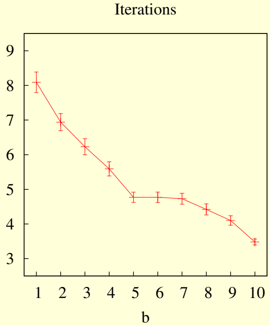
Batch size	Depth of TBT	Bound on number of iterations
1	2	2^n
2	3	1.7321^n
3	5	1.7100^n
4	8	1.6818^n
5	13	1.6703^n
6	21	1.6611^n
7	33	1.6479^n

Depth of TBT for batch size 7 due to Gerencsér *et al.* [GHDJ15].

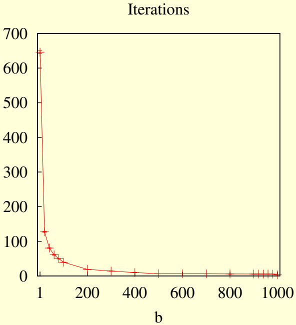
Will the bound continue to be non-increasing in the batch size?

If so, 1.6479^n would be a bound for Howard's Policy Iteration!

BSPI: Effect of Batch Size b



$n = 10$



$n = 1000$

Averaged over n-state, 2-action MDPs with randomly generated transition and reward functions. Each point is an average over 100 randomly-generated MDP instances and initial policies [KMG16a].

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

3. Review of MDP planning

Open Problems

- Is the complexity of Howard's PI on 2-action MDPs upper-bounded by the Fibonacci sequence ($\approx 1.6181^n$)?
- Is Howard's PI the most efficient among deterministic PI algorithms (worst case over all MDPs)?
- Is there a super-linear lower bound on the number of iterations taken by Howard's PI on 2-action MDPs?
- Is Howard's PI strongly polynomial on deterministic MDPs?
- Is there a variant of PI that can visit all k^n policies in some n-state, k-action MDP—implying an $\Omega(k^n)$ lower bound?
- Is there a strongly polynomial algorithm for MDP planning?

Markov Decision Problems

1. Policy iteration: variants and complexity bounds

2. Analysis of bounds
   - Basic tools
   - Howard's PI with $k = 2$
   - BSPI with $k = 2$
   - Open problems

3. Review of MDP planning

Summary of MDP Planning

- MDPs are an abstraction of sequential decision making.
- Many applications; many different formulations.
- Essential solution concept: optimal policy (known to exist).

- Three main families of planning algorithms: value iteration, linear programming, policy iteration.
- Have strengths and weaknesses in theory and in practice. Can combine.

- We showed correctness of all three methods.
- Used Banach's fixed-point theorem, Bellman (optimality) operator.

- What if T, R were not given, but have to be *learned* from interaction? Can we still learn to act optimally?
- Yes: that's the reinforcement learning problem. Next week!