Improved Bounds for Policy Iteration in Markov Decision Problems

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
shivaram@cse.iitb.ac.in

November 2017

Collaborators: Neeldhara Misra, Aditya Gopalan, Utkarsh Mall, Ritish Goyal, Anchit Gupta

Improved Bounds for Policy Iteration in Markov Decision Problems

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
shivaram@cse.iitb.ac.in

November 2017

Collaborators: Neeldhara Misra, Aditya Gopalan, Utkarsh Mall, Ritish Goyal, Anchit Gupta

Improved Bounds for Policy Iteration in Markov Decision Problems

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
shivaram@cse.iitb.ac.in

November 2017

Collaborators: Neeldhara Misra, Aditya Gopalan, Utkarsh Mall, Ritish Goyal, Anchit Gupta

Improved Bounds for Policy Iteration in Markov Decision Problems

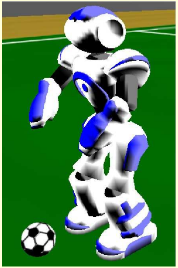
Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
shivaram@cse.iitb.ac.in

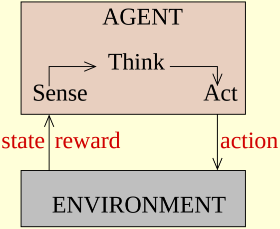
November 2017

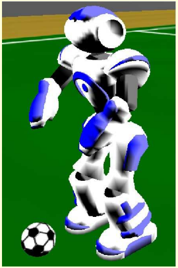
Collaborators: Neeldhara Misra, Aditya Gopalan, Utkarsh Mall, Ritish Goyal, Anchit Gupta

Sequential Decision Making

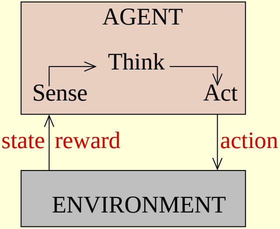


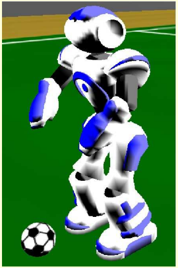
Sequential Decision Making





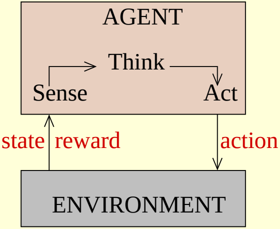
Sequential Decision Making

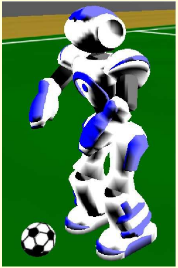




https://img.tradeindia.com/fp/1/524/panoramic-elevators-564.jpg

Sequential Decision Making

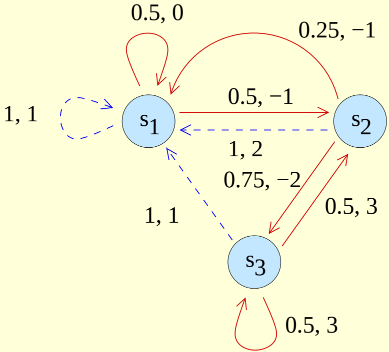




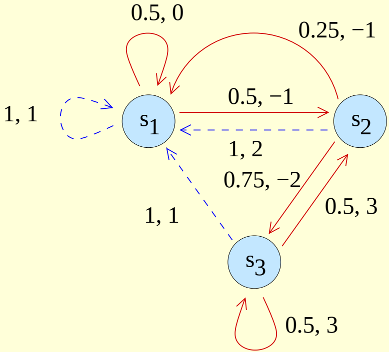
https://img.tradeindia.com/fp/1/524/panoramic-elevators-564.jpg

http://www.nature.com/polopoly_fs/7.33483.1453824868!/image/WEB_Go—1.jpg_gen/derivatives/landscape_630/WEB_Go—1.jpg

Markov Decision Problems (MDPs)



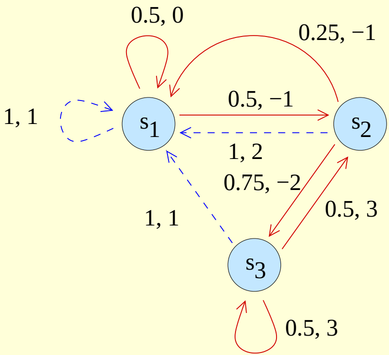
Markov Decision Problems (MDPs)



Elements of an MDP

- States (S)
- Actions (A)
- Transition probabilities (T)
- Rewards (R)

Markov Decision Problems (MDPs)



Elements of an MDP

States (S)

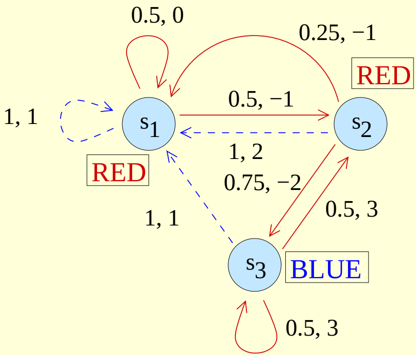
Actions (A)

Transition probabilities (T)

Rewards (R)

Behaviour is encoded as a **Policy** π, which maps states to actions.

Markov Decision Problems (MDPs)

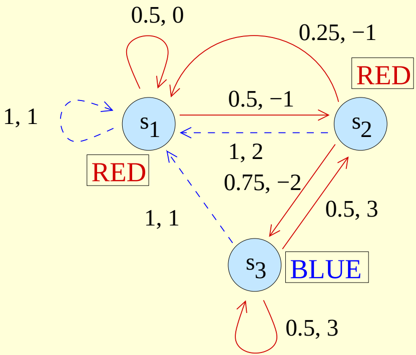


Elements of an MDP

- States (*S*)
- Actions (*A*)
- Transition probabilities (*T*)
- Rewards (*R*)

Behaviour is encoded as a **Policy** π, which maps states to actions.

Markov Decision Problems (MDPs)

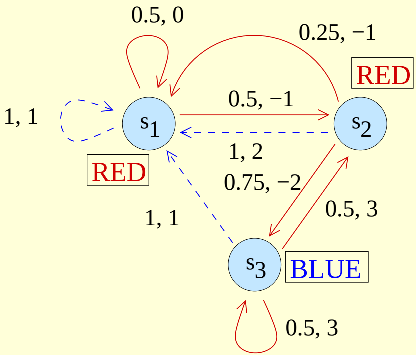


Elements of an MDP

- States (S)
- Actions (A)
- Transition probabilities (T)
- Rewards (R)

Behaviour is encoded as a **Policy** π, which maps states to actions.
What is a "good" policy?

Markov Decision Problems (MDPs)

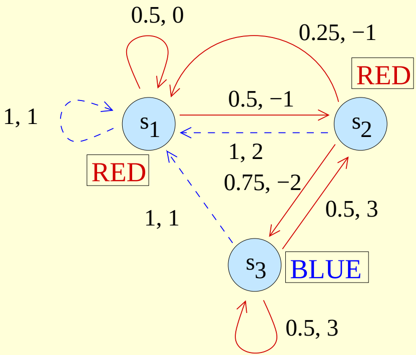


Elements of an MDP

- States (S)
- Actions (A)
- Transition probabilities (T)
- Rewards (R)

Behaviour is encoded as a **Policy** π, which maps states to actions.
What is a "good" policy? One that maximises expected long-term reward.

Markov Decision Problems (MDPs)



Elements of an MDP

States (S)

Actions (A)

Transition probabilities (T)

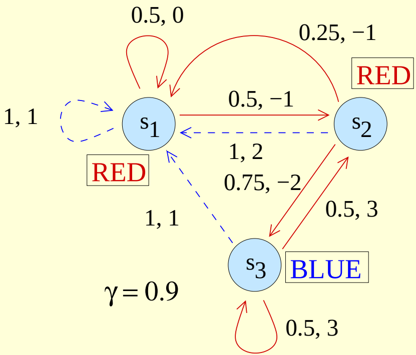
Rewards (R)

Behaviour is encoded as a **Policy** π, which maps states to actions.
What is a "good" policy? One that maximises expected long-term reward.

V^π is the **Value Function** of π. For $s \in S$,

$$V^\pi(s) = \mathbb{E}_\pi \left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \middle| \text{start state} = s \right].$$

Markov Decision Problems (MDPs)



Elements of an MDP

- States (S)
- Actions (A)
- Transition probabilities (T)
- Rewards (R)
- Discount factor (γ)

Behaviour is encoded as a **Policy** π, which maps states to actions.
What is a "good" policy? One that maximises expected long-term reward.

V^{π} is the **Value Function** of π. For $s \in S$,

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots | \text{start state} = s\right].$$

Optimal Policies

V^π satisfies a recursive equation: $V^\pi = R_\pi + \gamma T_\pi V^\pi$, which gives $V^\pi = (I - \gamma T_\pi)^{-1} R_\pi$.

Optimal Policies

V^π satisfies a recursive equation: $V^\pi = R_\pi + \gamma T_\pi V^\pi$, which gives $V^\pi = (I - \gamma T_\pi)^{-1} R_\pi$.

π	$V^\pi(s_1)$	$V^\pi(s_2)$	$V^\pi(s_3)$
RRR	4.45	6.55	10.82
RRB	-5.61	-5.75	-4.05
RBR	2.76	4.48	9.12
RBB	2.76	4.48	3.48
BRR	10.0	9.34	13.10
BRB	10.0	7.25	10.0
BBR	10.0	11.0	14.45
BBB	10.0	11.0	10.0

Optimal Policies

V^π satisfies a recursive equation: $V^\pi = R_\pi + \gamma T_\pi V^\pi$, which gives $V^\pi = (I - \gamma T_\pi)^{-1} R_\pi$.

π	$V^\pi(s_1)$	$V^\pi(s_2)$	$V^\pi(s_3)$	
RRR	4.45	6.55	10.82	
RRB	-5.61	-5.75	-4.05	
RBR	2.76	4.48	9.12	
RBB	2.76	4.48	3.48	
BRR	10.0	9.34	13.10	
BRB	10.0	7.25	10.0	
BBR	**10.0**	**11.0**	**14.45**	$\leftarrow$ Optimal policy
BBB	10.0	11.0	10.0	

Optimal Policies

V^π satisfies a recursive equation: $V^\pi = R_\pi + \gamma T_\pi V^\pi$, which gives $V^\pi = (I - \gamma T_\pi)^{-1} R_\pi$.

π	$V^\pi(s_1)$	$V^\pi(s_2)$	$V^\pi(s_3)$	
RRR	4.45	6.55	10.82	
RRB	-5.61	-5.75	-4.05	
RBR	2.76	4.48	9.12	
RBB	2.76	4.48	3.48	
BRR	10.0	9.34	13.10	
BRB	10.0	7.25	10.0	
BBR	**10.0**	**11.0**	**14.45**	← Optimal policy
BBB	10.0	11.0	10.0	

Every MDP is guaranteed to have an optimal policy $\pi^\star$, such that

$$\forall \pi \in \Pi, \forall s \in S : V^{\pi^\star}(s) \geq V^\pi(s).$$

Optimal Policies

V^{π} satisfies a recursive equation: $V^{\pi} = R_{\pi} + \gamma T_{\pi} V^{\pi}$, which gives $V^{\pi} = (I - \gamma T_{\pi})^{-1} R_{\pi}$.

π	$V^{\pi}(s_1)$	$V^{\pi}(s_2)$	$V^{\pi}(s_3)$	
RRR	4.45	6.55	10.82	
RRB	-5.61	-5.75	-4.05	
RBR	2.76	4.48	9.12	
RBB	2.76	4.48	3.48	
BRR	10.0	9.34	13.10	
BRB	10.0	7.25	10.0	
BBR	**10.0**	**11.0**	**14.45**	← Optimal policy
BBB	10.0	11.0	10.0	

Every MDP is guaranteed to have an optimal policy $\pi^{\star}$, such that

$$\forall \pi \in \Pi, \forall s \in S : V^{\pi^{\star}}(s) \geq V^{\pi}(s).$$

What is the complexity of computing an optimal policy?

Optimal Policies

V^π satisfies a recursive equation: $V^\pi = R_\pi + \gamma T_\pi V^\pi$, which gives $V^\pi = (I - \gamma T_\pi)^{-1} R_\pi$.

π	$V^\pi(s_1)$	$V^\pi(s_2)$	$V^\pi(s_3)$	
RRR	4.45	6.55	10.82	
RRB	-5.61	-5.75	-4.05	
RBR	2.76	4.48	9.12	
RBB	2.76	4.48	3.48	
BRR	10.0	9.34	13.10	
BRB	10.0	7.25	10.0	
BBR	**10.0**	**11.0**	**14.45**	$\leftarrow$ Optimal policy
BBB	10.0	11.0	10.0	

Every MDP is guaranteed to have an optimal policy $\pi^\star$, such that

$$\forall \pi \in \Pi, \forall s \in S : V^{\pi^\star}(s) \geq V^\pi(s).$$

What is the complexity of computing an optimal policy?

Note: an MDP with $|S| = n$ states and $|A| = k$ actions has a total of k^n policies.

Optimal Policies

V^{π} satisfies a recursive equation: $V^{\pi} = R_{\pi} + \gamma T_{\pi} V^{\pi}$, which gives $V^{\pi} = (I - \gamma T_{\pi})^{-1} R_{\pi}$.

π	$V^{\pi}(s_1)$	$V^{\pi}(s_2)$	$V^{\pi}(s_3)$	
RRR	4.45	6.55	10.82	
RRB	-5.61	-5.75	-4.05	
RBR	2.76	4.48	9.12	
RBB	2.76	4.48	3.48	
BRR	10.0	9.34	13.10	
BRB	10.0	7.25	10.0	
BBR	**10.0**	**11.0**	**14.45**	$\leftarrow$ Optimal policy
BBB	10.0	11.0	10.0	

Every MDP is guaranteed to have an optimal policy $\pi^{\star}$, such that

$$\forall \pi \in \Pi, \forall s \in S : V^{\pi^{\star}}(s) \geq V^{\pi}(s).$$

What is the complexity of computing an optimal policy?

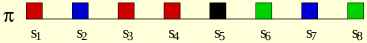
Note: an MDP with $|S| = n$ states and $|A| = k$ actions has a total of k^n policies.

One extra definition needed: **Action Value Function** Q^{π}_a for $a \in A$.

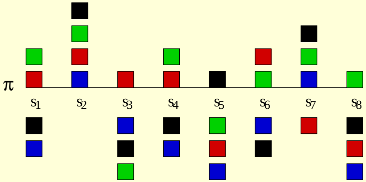
$Q^{\pi}_a = R_a + \gamma T_a V^{\pi}$.

Given π, a polynomial computation yields V^{π} and Q^{π}_a for $a \in A$.

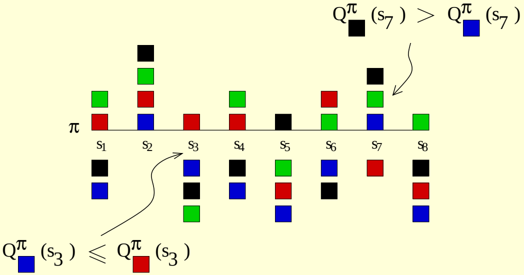
Policy Improvement



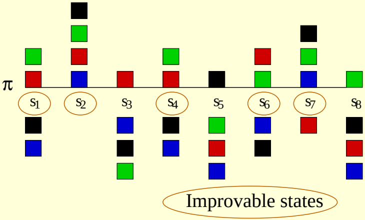
Policy Improvement



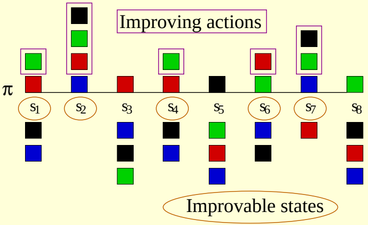
Policy Improvement



Policy Improvement



Policy Improvement



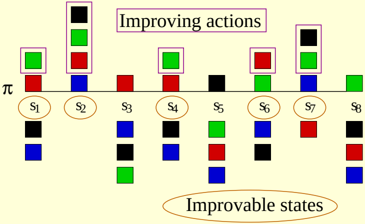
Policy Improvement

Given π,

Pick one or more improvable states, and in them,

Switch to an arbitrary improving action.

Let the resulting policy be π'.



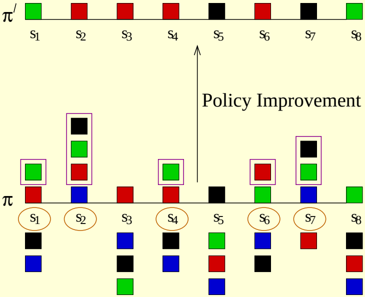
Policy Improvement

Given π,

  Pick one or more improvable states, and in them,

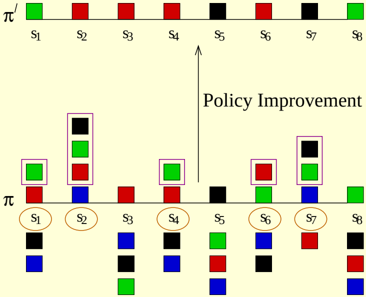
  Switch to an arbitrary improving action.

Let the resulting policy be π'.



Policy Improvement

Given π,
  Pick one or more improvable states, and in them,
  Switch to an arbitrary improving action.
Let the resulting policy be π'.



Policy Improvement Theorem (H60, B12):
(1) If π has no improvable states, then it is optimal, else
(2) if π' is obtained as above, then

$$\forall s \in S : V^{\pi'}(s) \geq V^{\pi}(s) \text{ and } \exists s \in S : V^{\pi'}(s) > V^{\pi}(s).$$

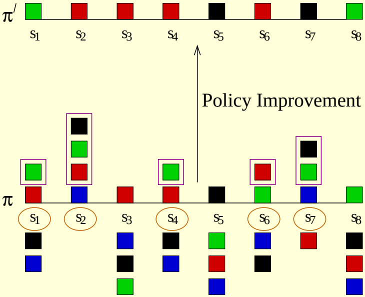
Policy Improvement

Given π,

Pick one or more improvable states, and in them,

Switch to an arbitrary improving action.

Let the resulting policy be π'.



Policy Improvement Theorem (H60, B12):

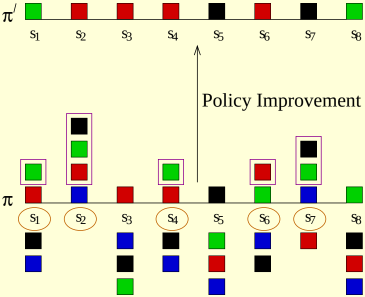
(1) If π has no improvable states, then it is optimal, else

(2) if π' is obtained as above, then

$$\forall s \in S : V^{\pi'}(s) \geq V^{\pi}(s) \text{ and } \exists s \in S : V^{\pi'}(s) > V^{\pi}(s).$$

Policy Improvement

Given π,
Pick one or more improvable states, and in them,
Switch to an arbitrary improving action.
Let the resulting policy be π'.



Policy Improvement Theorem (H60, B12):
(1) If π has no improvable states, then it is optimal, else
(2) if π' is obtained as above, then

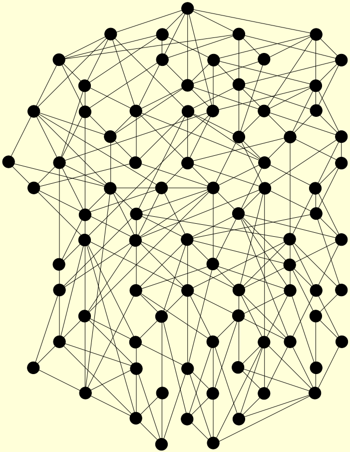
$$\forall s \in S : V^{\pi'}(s) \geq V^{\pi}(s) \text{ and } \exists s \in S : V^{\pi'}(s) > V^{\pi}(s).$$

Policy Iteration (PI)

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

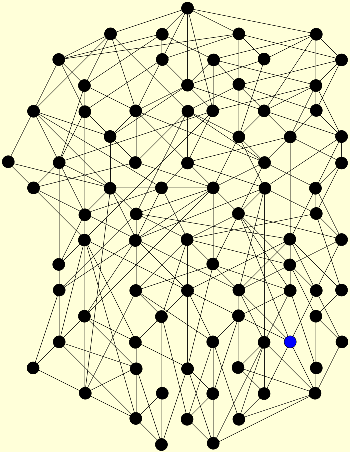
Policy Iteration (PI)

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



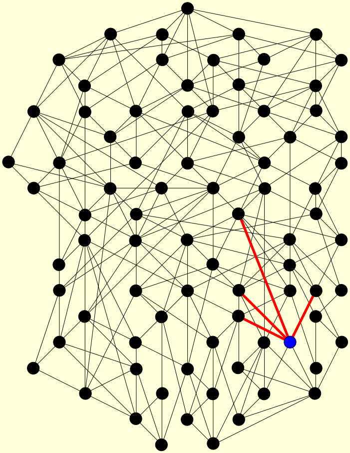
Policy Iteration (PI)

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



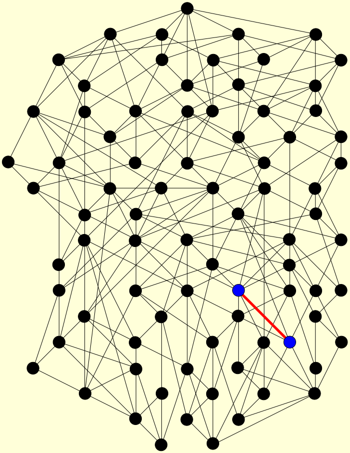
Policy Iteration (PI)

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



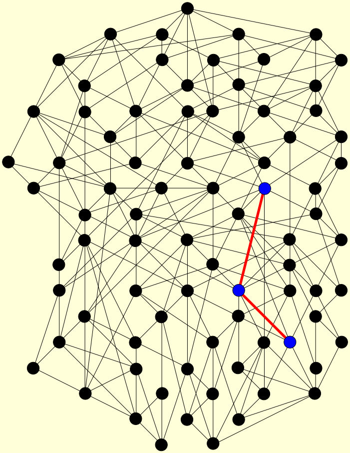
Policy Iteration (PI)

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



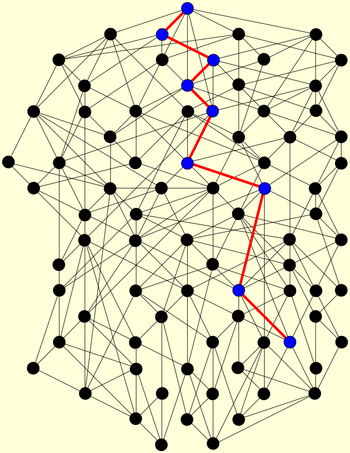
Policy Iteration (PI)

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



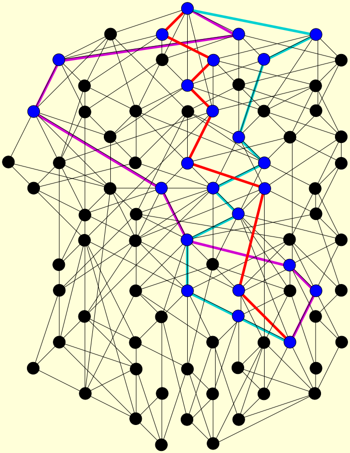
Policy Iteration (PI)

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



Policy Iteration (PI)

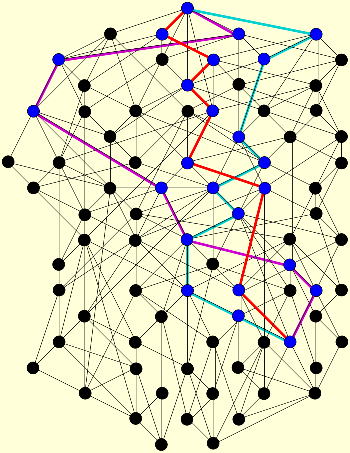
$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi \leftarrow$ PolicyImprovement(π).



Different switching strategies lead to different routes to the top.

Policy Iteration (PI)

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



Different switching strategies lead to different routes to the top.
How long are the routes?!

Switching Strategies and Bounds

Upper bounds on number of iterations

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

Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	Type	$k = 2$	General k
Howard's PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Randomised PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$

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 PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Randomised PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$

Lower bounds on number of iterations

$\Omega(n)$ Howard's PI on n-state, 2-action MDPs [HZ10].
$\Omega(1.4142^n)$ Simple PI on n-state, 2-action MDPs [MC94].

Switching Strategies and Bounds

Upper bounds on number of iterations

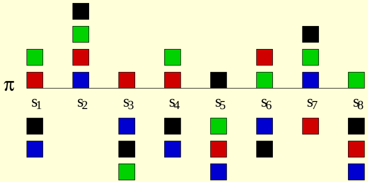
PI Variant	**Type**	$k = 2$	**General** k
Howard's PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Randomised PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$
Batch-switching PI [KMG16a, GK17]	Deterministic	1.6479^n	$k^{0.7207n}$
Recursive Simple PI [KMG16b]	Randomised	–	$(2 + \ln(k-1))^n$

Lower bounds on number of iterations

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

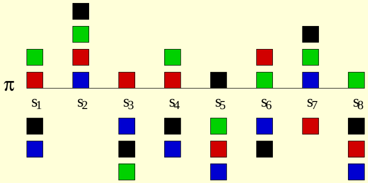
$\Omega(1.4142^n)$ Simple PI on n-state, 2-action MDPs [MC94].

Recursive Simple Policy Iteration



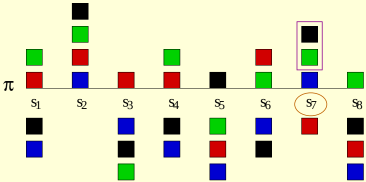
Recursive Simple Policy Iteration

Given π,
 Pick the improvable state with the highest index, and,
 Switch to an improving action picked uniformly at random.
Let the resulting policy be π'.



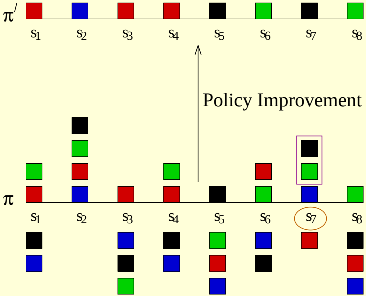
Recursive Simple Policy Iteration

Given π,
  Pick the improvable state with the highest index, and,
  Switch to an improving action picked uniformly at random.
Let the resulting policy be π'.



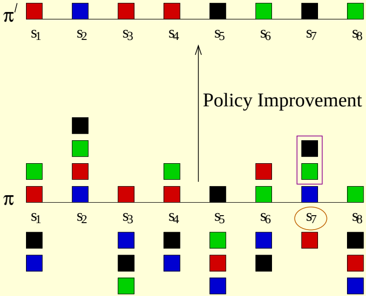
Recursive Simple Policy Iteration

Given π,
  Pick the improvable state with the highest index, and,
  Switch to an improving action picked uniformly at random.
Let the resulting policy be π'.



Recursive Simple Policy Iteration

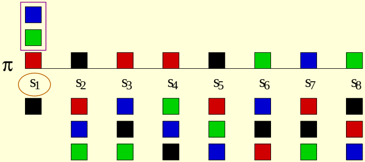
Given π,
  Pick the improvable state with the highest index, and,
  Switch to an improving action picked uniformly at random.
Let the resulting policy be π'.



Expected number of iterations: $(1 + H_{k-1})^n \leq (2 + \ln(k-1))^n$.

Recursive Simple Policy Iteration

Given π,
Pick the improvable state with the highest index, and,
Switch to an improving action picked uniformly at random.
Let the resulting policy be π'.



Expected number of iterations: $(1 + H_{k-1})^n \leq (2 + \ln(k-1))^n$.

Conclusion

Policy Iteration: widely used algorithm, more than half a century old.

Substantial gap exists between upper and lower bounds.

We furnish several exponential improvements to upper bounds.

Conclusion

Policy Iteration: widely used algorithm, more than half a century old.

Substantial gap exists between upper and lower bounds.

We furnish several exponential improvements to upper bounds.

Bears similarity to Simplex algorithm for Linear Programming.

Howard's PI works much better in practice than the variants for which we have shown improved upper bounds!

Open problem: Is the number of iterations taken by Howard's PI on n-state, 2-action MDPs upper-bounded by the $(n+2)$-nd Fibonacci number?

Conclusion

Policy Iteration: widely used algorithm, more than half a century old.

Substantial gap exists between upper and lower bounds.

We furnish several exponential improvements to upper bounds.

Bears similarity to Simplex algorithm for Linear Programming.

Howard's PI works much better in practice than the variants for which we have shown improved upper bounds!

Open problem: Is the number of iterations taken by Howard's PI on n-state, 2-action MDPs upper-bounded by the $(n+2)$-nd Fibonacci number?

For references see tutorial.

Theoretical Analysis of Policy Iteration
Tutorial at IJCAI 2017
https://www.cse.iitb.ac.in/ shivaram/resources/ijcai-2017-tutorial-policyiteration/index.html.

Conclusion

Policy Iteration: widely used algorithm, more than half a century old.

Substantial gap exists between upper and lower bounds.

We furnish several exponential improvements to upper bounds.

Bears similarity to Simplex algorithm for Linear Programming.

Howard's PI works much better in practice than the variants for which we have shown improved upper bounds!

Open problem: Is the number of iterations taken by Howard's PI on n-state, 2-action MDPs upper-bounded by the $(n+2)$-nd Fibonacci number?

For references see tutorial.

> Theoretical Analysis of Policy Iteration
> Tutorial at IJCAI 2017
> https://www.cse.iitb.ac.in/ shivaram/resources/ijcai-2017-tutorial-policyiteration/index.html.

Thank you!