Algorithms for MDP Planning

Shivaram Kalyanakrishnan

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

August 2018

Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms

Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms

Value Iteration

$V_0 \leftarrow$ Arbitrary, element-wise bounded, n-length vector. $t \leftarrow 0$.
Repeat:
For $s \in S$**:**
$V_{t+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V_t(s')\right)$.
$t \leftarrow t + 1$.
Until $V_t \approx V_{t-1}$ (up to machine precision).

Value Iteration

$V_0 \leftarrow$ Arbitrary, element-wise bounded, n-length vector. $t \leftarrow 0$.
Repeat:
  For $s \in S$**:**
    $V_{t+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V_t(s')\right)$.
    $t \leftarrow t + 1$.
Until $V_t \approx V_{t-1}$ (up to machine precision).

Convergence to $V^\star$ guaranteed using a max-norm contraction argument.

Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms

Linear Programming

$$\text{Minimise} \quad \sum_{s \in S} V(s)$$

$$\text{subject to} \quad V(s) \geq \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V(s')\right), \forall s \in S, \forall a \in A.$$

Linear Programming

$$\text{Minimise} \quad \sum_{s \in S} V(s)$$

$$\text{subject to} \quad V(s) \geq \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V(s')\right), \forall s \in S, \forall a \in A.$$

Let $|S| = n$ and $|A| = k$.

Linear Programming

$$\text{Minimise} \quad \sum_{s \in S} V(s)$$

$$\text{subject to} \quad V(s) \geq \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V(s') \right), \forall s \in S, \forall a \in A.$$

Let $|S| = n$ and $|A| = k$.

n variables, nk constraints.

Linear Programming

Minimise $\sum_{s \in S} V(s)$

subject to $V(s) \geq \sum_{s' \in S} T(s, a, s') \left(R(s, a, s') + \gamma V(s') \right), \forall s \in S, \forall a \in A.$

Let $|S| = n$ and $|A| = k$.

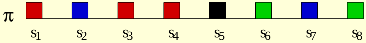
n variables, nk constraints.

Can also be posed as *dual* with nk variables and n constraints.

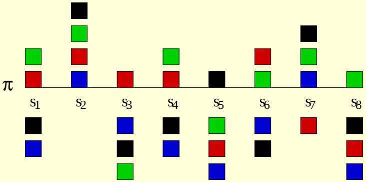
Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms

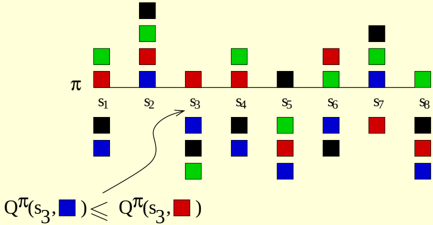
Policy Improvement



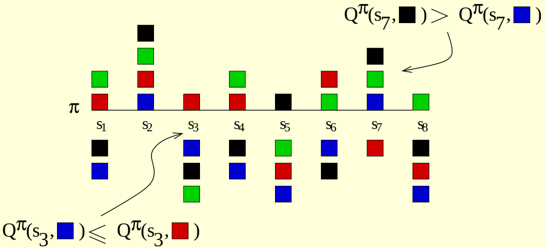
Policy Improvement



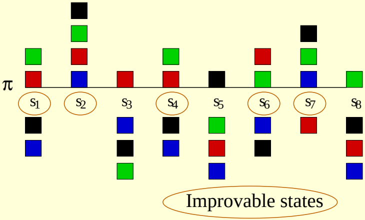
Policy Improvement



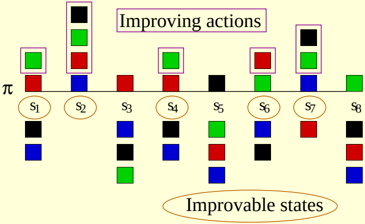
Policy Improvement



Policy Improvement

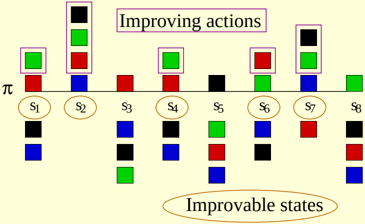


Policy Improvement



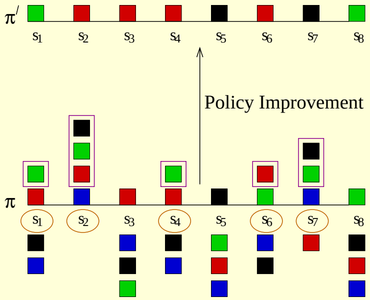
Policy Improvement

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



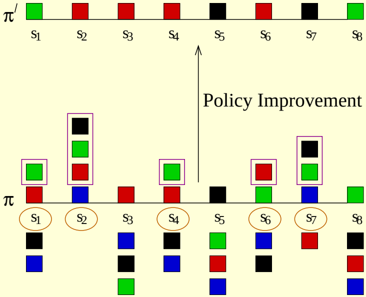
Policy Improvement

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

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

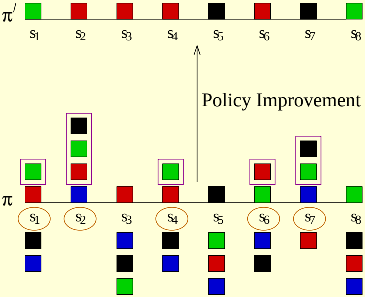
Policy Improvement

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:

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

(2) if π' is obtained as above, then

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

Definitions and Basic Facts

- For $X : S \to \mathbb{R}$ and $Y : S \to \mathbb{R}$, we define $X \succeq Y$ if $\forall s \in S : X(s) \geq Y(s)$, and we define $X \succ Y$ if $X \succeq Y$ and $\exists s \in S : X(s) > Y(s)$.

Definitions and Basic Facts

- For $X : S \to \mathbb{R}$ and $Y : S \to \mathbb{R}$, we define $X \succeq Y$ if $\forall s \in S : X(s) \geq Y(s)$, and we define $X \succ Y$ if $X \succeq Y$ and $\exists s \in S : X(s) > Y(s)$.

  For policies $\pi_1, \pi_2 \in \Pi$, we define $\pi_1 \succeq \pi_2$ if $V^{\pi_1} \succeq V^{\pi_2}$, and we define $\pi_1 \succ \pi_2$ if $V^{\pi_1} \succ V^{\pi_2}$.

Definitions and Basic Facts

- For $X : S \to \mathbb{R}$ and $Y : S \to \mathbb{R}$, we define $X \succeq Y$ if $\forall s \in S : X(s) \geq Y(s)$, and we define $X \succ Y$ if $X \succeq Y$ and $\exists s \in S : X(s) > Y(s)$.

  For policies $\pi_1, \pi_2 \in \Pi$, we define $\pi_1 \succeq \pi_2$ if $V^{\pi_1} \succeq V^{\pi_2}$, and we define $\pi_1 \succ \pi_2$ if $V^{\pi_1} \succ V^{\pi_2}$.

- **Bellman Operator.** For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows: for $X : S \to \mathbb{R}$ and $\forall s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s')\right).$$

Definitions and Basic Facts

- For $X : S \to \mathbb{R}$ and $Y : S \to \mathbb{R}$, we define $X \succeq Y$ if $\forall s \in S : X(s) \geq Y(s)$, and we define $X \succ Y$ if $X \succeq Y$ and $\exists s \in S : X(s) > Y(s)$.

  For policies $\pi_1, \pi_2 \in \Pi$, we define $\pi_1 \succeq \pi_2$ if $V^{\pi_1} \succeq V^{\pi_2}$, and we define $\pi_1 \succ \pi_2$ if $V^{\pi_1} \succ V^{\pi_2}$.

- **Bellman Operator.** For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows: for $X : S \to \mathbb{R}$ and $\forall s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s') \right).$$

- **Fact 1**. For $\pi \in \Pi$, $X : S \to \mathbb{R}$, and $Y : S \to \mathbb{R}$:

$$\text{if } X \succeq Y, \text{ then } B^\pi(X) \succeq B^\pi(Y).$$

Definitions and Basic Facts

- For $X : S \to \mathbb{R}$ and $Y : S \to \mathbb{R}$, we define $X \succeq Y$ if $\forall s \in S : X(s) \geq Y(s)$, and we define $X \succ Y$ if $X \succeq Y$ and $\exists s \in S : X(s) > Y(s)$.

  For policies $\pi_1, \pi_2 \in \Pi$, we define $\pi_1 \succeq \pi_2$ if $V^{\pi_1} \succeq V^{\pi_2}$, and we define $\pi_1 \succ \pi_2$ if $V^{\pi_1} \succ V^{\pi_2}$.

- **Bellman Operator.** For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows: for $X : S \to \mathbb{R}$ and $\forall s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s')\right).$$

- **Fact 1**. For $\pi \in \Pi$, $X : S \to \mathbb{R}$, and $Y : S \to \mathbb{R}$:

$$\text{if } X \succeq Y, \text{ then } B^\pi(X) \succeq B^\pi(Y).$$

- **Fact 2**. For $\pi \in \Pi$ and $X : S \to \mathbb{R}$:

$$\lim_{l \to \infty} (B^\pi)^l(X) = V^\pi. \text{ (from Banach's FP Theorem)}$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty}(B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty}(B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

π has improvable states and policy improvement yields π'

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

π has improvable states and policy improvement yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

π has improvable states and policy improvement yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

π has improvable states and policy improvement yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi}) \succeq \cdots \succeq (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

Observe that for $\pi, \pi' \in \Pi, \forall s \in S$: $B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))$.

π has no improvable states

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

π has improvable states and policy improvement yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi}) \succeq \cdots \succeq (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

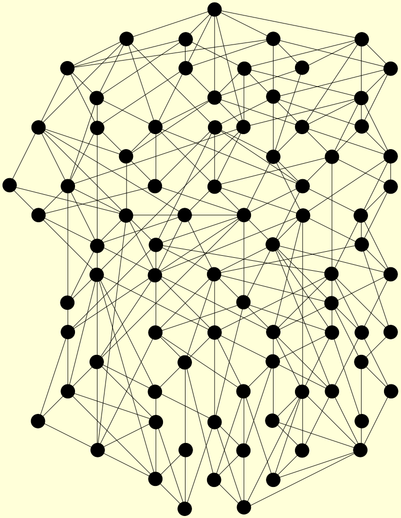
$$\implies V^{\pi'} \succ V^{\pi}.$$

Policy Iteration Algorithm

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

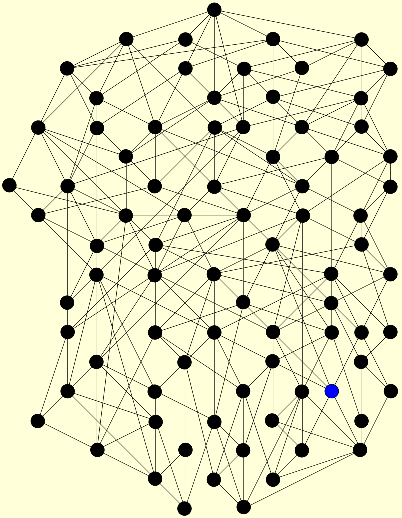
Policy Iteration Algorithm

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



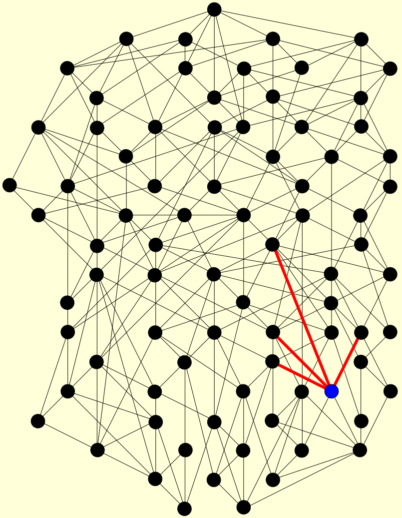
Policy Iteration Algorithm

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



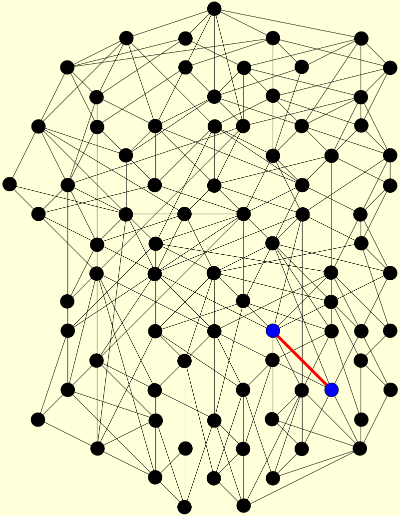
Policy Iteration Algorithm

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



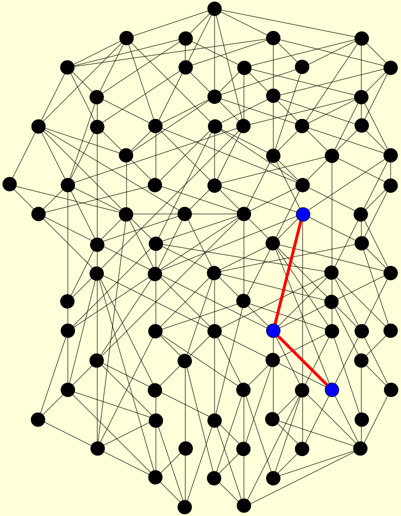
Policy Iteration Algorithm

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



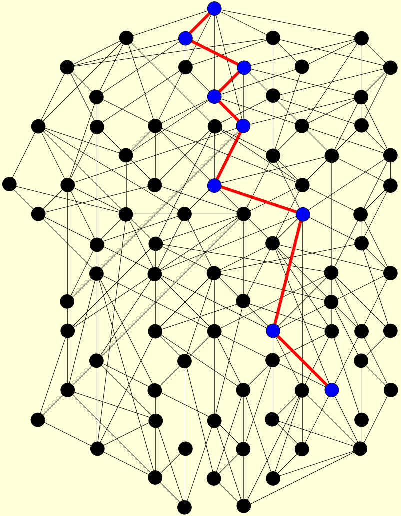
Policy Iteration Algorithm

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



Policy Iteration Algorithm

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



Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms

Overview

1. Value Iteration
2. Linear Programming
3. Policy Iteration
   - Policy Improvement Theorem
4. Complexity of algorithms (not a part of course syllabus!)

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly($n, k, B, \frac{1}{1-\gamma}$), where B is the number of bits used to represent the MDP.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly$(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly($n, k, B, \frac{1}{1-\gamma}$), where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly($n, k, B, \frac{1}{1-\gamma}$), where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*?

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly(n, k, B, $\frac{1}{1-\gamma}$), where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2k + n^3)$.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.
- Upper Bound for Value Iteration [LDK95]:
  poly($n, k, B, \frac{1}{1-\gamma}$), where B is the number of bits used to represent the MDP.
  Not a strong bound.
- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2k + n^3)$.
  Can you give a strong bound on the running time of MDP planning?

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly$(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2k + n^3)$.
  Can you give a strong bound on the running time of MDP planning? poly$(n, k) \cdot k^n$.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly$(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2k + n^3)$.
  Can you give a strong bound on the running time of MDP planning? poly$(n, k) \cdot k^n$.

- Bounds for Linear Programming-type approaches to MDP planning:
  poly(n, k, B) [K80, K84].
  poly$(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].
  poly$(n, k) \cdot k^{0.6834n}$ [GK17].

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  $\text{poly}(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2 k + n^3)$.
  Can you give a strong bound on the running time of MDP planning? $\text{poly}(n, k) \cdot k^n$.

- Bounds for Linear Programming-type approaches to MDP planning:
  $\text{poly}(n, k, B)$ [K80, K84].
  $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].
  $\text{poly}(n, k) \cdot k^{0.6834n}$ [GK17].
  $\text{poly}(n, k)$ for deterministic MDPs [MTZ10, PY13].

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  poly$(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2k + n^3)$.
  Can you give a strong bound on the running time of MDP planning? poly$(n, k) \cdot k^n$.

- Bounds for Linear Programming-type approaches to MDP planning:
  poly(n, k, B) [K80, K84].
  poly$(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].
  poly$(n, k) \cdot k^{0.6834n}$ [GK17].
  poly(n, k) for deterministic MDPs [MTZ10, PY13].

- Complexity of Policy Iteration trivially upper-bounded by poly$(n, k) \cdot k^n$.

Weak and Strong Running-time Bounds

- Computation model: Infinite precision arithmetic (or Real RAM) model.

- Upper Bound for Value Iteration [LDK95]:
  $\text{poly}(n, k, B, \frac{1}{1-\gamma})$, where B is the number of bits used to represent the MDP.
  Not a strong bound.

- Strong bounds depend solely on n and k (no dependence on B, γ, etc.).
  Is there a strong upper bound on the complexity of *policy evaluation*? $O(n^2 k + n^3)$.
  Can you give a strong bound on the running time of MDP planning? $\text{poly}(n, k) \cdot k^n$.

- Bounds for Linear Programming-type approaches to MDP planning:
  $\text{poly}(n, k, B)$ [K80, K84].
  $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].
  $\text{poly}(n, k) \cdot k^{0.6834n}$ [GK17].
  $\text{poly}(n, k)$ for deterministic MDPs [MTZ10, PY13].

- Complexity of Policy Iteration trivially upper-bounded by $\text{poly}(n, k) \cdot k^n$.
  Is it more efficient than that?

Switching Strategies and Bounds for Policy Iteration

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's ("all switch") 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	$\approx O\left(\left(\frac{k}{2}\right)^n\right)$

Switching Strategies and Bounds for Policy Iteration

Upper bounds on number of iterations

PI Variant	Type	$k = 2$	General k
Howard's ("all switch") 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	$\approx O\left(\left(\frac{k}{2}\right)^n\right)$
Batch-switching PI (BSPI) [KMG16a]	Deterministic	1.6479^n	–
Recursive BSPI [GK17]	Deterministic	–	$k^{0.7207n}$
Recursive Simple PI [KMG16b]	Randomised	–	$(2 + \ln(k-1))^n$

Switching Strategies and Bounds for Policy Iteration

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's ("all switch") 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	$\approx O\left(\left(\frac{k}{2}\right)^n\right)$
Batch-switching PI (BSPI) [KMG16a]	Deterministic	1.6479^n	–
Recursive BSPI [GK17]	Deterministic	–	$k^{0.7207n}$
Recursive Simple PI [KMG16b]	Randomised	–	$(2 + \ln(k-1))^n$

Lower bounds on number of iterations

$\Omega(2^{n/7})$ Howard's PI on n-state MDPs with $\Theta(n)$ actions per state [F10, HGD12].

Switching Strategies and Bounds for Policy Iteration

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's ("all switch") 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	$\approx O\left(\left(\frac{k}{2}\right)^n\right)$
Batch-switching PI (BSPI) [KMG16a]	Deterministic	1.6479^n	–
Recursive BSPI [GK17]	Deterministic	–	$k^{0.7207n}$
Recursive Simple PI [KMG16b]	Randomised	–	$(2 + \ln(k-1))^n$

Lower bounds on number of iterations

$\Omega(2^{n/7})$ Howard's PI on n-state MDPs with $\Theta(n)$ actions per state [F10, HGD12].
$\Omega(2^{n/2})$ Simple PI on n-state, 2-action MDPs [MC94].

Switching Strategies and Bounds for Policy Iteration

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's ("all switch") 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	$\approx O\left(\left(\frac{k}{2}\right)^n\right)$
Batch-switching PI (BSPI) [KMG16a]	Deterministic	1.6479^n	–
Recursive BSPI [GK17]	Deterministic	–	$k^{0.7207n}$
Recursive Simple PI [KMG16b]	Randomised	–	$(2 + \ln(k - 1))^n$

Lower bounds on number of iterations

$\Omega(2^{n/7})$ Howard's PI on n-state MDPs with $\Theta(n)$ actions per state [F10, HGD12].

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

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

Open Problems

- Is the complexity of Howard's PI on 2-action MDPs upper-bounded by the Fibonacci sequence ($\approx 1.6181^n$)?

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)?

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 iterations taken by Howard's PI on 2-action MDPs?

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 iterations taken by Howard's PI on 2-action MDPs?
- Is (Howard's) PI strongly polynomial on deterministic MDPs?

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 iterations taken by Howard's PI on 2-action MDPs?
- Is (Howard's) PI strongly polynomial on deterministic MDPs?
- Is there a strongly polynomial algorithm for MDP planning?

References and Additional Reading

R. A. Howard, 1960. Dynamic Programming and Markov Processes. MIT Press, 1960.

L. G. Khachiyan, 1980. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72.

N. Karmarkar, 1984. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.

Mary Melekopoglou and Anne Condon, 1994. On the complexity of the policy improvement algorithm for Markov decision processes. *INFORMS Journal on Computing*, 6(2):188–192, 1994.

Martin L. Puterman, 1994. Markov Decision Processes. Wiley, 1994.

Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling, 1995. On the complexity of solving Markov decision problems. *In Proc. UAI 1995*, pp. 394–402, Morgan Kaufmann, 1995.

Jiří Matoušek, Micha Sharir, and Emo Welzl, 1996. A Subexponential Bound for Linear Programming. *Algorithmica*, 16(4/5):498–516, 1996.

Yishay Mansour and Satinder Singh, 1999. On the Complexity of Policy Iteration. *In Proc. UAI 1999*, pp. 401–408, AUAI, 1999.

Daniel A. Spielman and Shang-Hua Teng, 2004. *Journal of the ACM*, 51(3):385–463, 2004.

John Fearnley, 2010. Exponential Lower Bounds for Policy Iteration. *In Proc. ICALP 2010*, pp. 551–562, Springer, 2010.

Thomas Dueholm Hansen and Uri Zwick, 2010. Lower bounds for Howard's algorithm for finding minimum mean-cost cycles. *In Proc. ISAAC 2010*, pp. 415–426, Springer 2010.

References and Additional Reading

Omid Madani, Mikkel Thorup, and Uri Zwick, 2010. Discounted deterministic Markov decision processes and discounted all-pairs shortest paths. *ACM Transactions on Algorithms*, 6(2):33:1–33:25, 2010.

Thomas Dueholm Hansen, 2012. Worst-case Analysis of Strategy Iteration and the Simplex Method. PhD thesis, Department of Computer Science, Aarhus University, July 2012.

Romain Hollanders, Balázs Gerencsér, Jean-Charles Delvenne, 2012. The complexity of policy iteration is exponential for discounted Markov decision processes. *In Proc. CDC 2012*, pp. 5997–6002, IEEE, 2012.

Ian Post and Yinyu Ye. The Simplex Method is Strongly Polynomial for Deterministic Markov Decision Processes. *In Proc. SODA 2013*, pp.1465–1473, SIAM, 2013.

Romain Hollanders, Balázs Gerencsér, Jean-Charles Delvenne, and Raphaël M. Jungers, 2014. Improved bound on the worst case complexity of policy iteration. `http://arxiv.org/pdf/1410.7583v1.pdf`.

Balázs Gerencsér, Romain Hollanders, Jean-Charles Delvenne, and Raphaël M. Jungers, 2015. A complexity analysis of policy iteration through combinatorial matrices arising from unique sink orientations.`http://arxiv.org/pdf/1407.4293v2.pdf`.

Shivaram Kalyanakrishnan, Utkarsh Mall, and Ritish Goyal, 2016a. Batch-Switching Policy Iteration. *In Proc. IJCAI 2016*, pp. 3147–3153, AAAI Press, 2016.

Shivaram Kalyanakrishnan, Neeldhara Misra, and Aditya Gopalan, 2016b. Randomised Procedures for Initialising and Switching Actions in Policy Iteration. *In Proc. AAAI 2016*, pp. 3145–3151, AAAI Press, 2016.

Anchit Gupta and Shivaram Kalyanakrishnan, 2017. Improved Strong Worst-case Upper Bounds for MDP Planning. *In Proc. IJCAI 2017*, pp. 1788–1794, IJCAI, 2017.