The Simplex and Policy-Iteration Methods are Strongly Polynomial for the Markov Decision Problem with Fixed Discount

Yinyu Ye
http://www.stanford.edu/~yyye

Department of Management Science and Engineering and
Institute of Computational and Mathematical Engineering
Stanford University

January 18-21, 2011

Outline

- The Markov Decision Process and its History
- The Simplex and Policy-Iteration Methods
- The Main Result: Strong Polynomiality
- Proof Sketch of the Main Result
- Remarks and Open Questions

The Markov Decision Process

- Markov decision processes (MDPs), named after Andrey Markov, provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.

The Markov Decision Process

- Markov decision processes (MDPs), named after Andrey Markov, provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.
- MDPs are useful for studying a wide range of optimization problems solved via dynamic programming, where it was known at least as early as the 1950s (cf. Shapley 1953, Bellman 1957).

The Markov Decision Process

- Markov decision processes (MDPs), named after Andrey Markov, provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.
- MDPs are useful for studying a wide range of optimization problems solved via dynamic programming, where it was known at least as early as the 1950s (cf. Shapley 1953, Bellman 1957).
- At each time step, the process is in some state i, and the decision maker choose an action $j \in \mathcal{A}_i$ that is available in state i. The process responds at the next time step by randomly moving into a new state i', and giving the decision maker a corresponding reward or cost $c^j(i, i')$.

The Markov Decision Process continued

- The probability that the process enters i' as its new state is influenced by the chosen state-action. Specifically, it is given by the state transition function $P^j(i, i')$. Thus, the next state i' depends on the current state i and the decision maker's action j.

The Markov Decision Process continued

- The probability that the process enters i' as its new state is influenced by the chosen state-action. Specifically, it is given by the state transition function $P^j(i, i')$. Thus, the next state i' depends on the current state i and the decision maker's action j.
- But given i and j, it is conditionally independent of all previous states and actions; in other words, the state transitions of an MDP possess the Markov property.

The Markov Decision Process continued

- A stationary policy for the decision maker is a set function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies the state-action π_i that the decision maker will choose when in state i. The MDP is to find a stationary policy to minimize the expected discounted sum over an infinite horizon:

$$\sum_{t=0}^{\infty} \gamma^t c^{\pi_{i^t}}(i^t, i^{t+1}),$$

where $0 \leq \gamma < 1$ is the so-called discount rate.

The Markov Decision Process continued

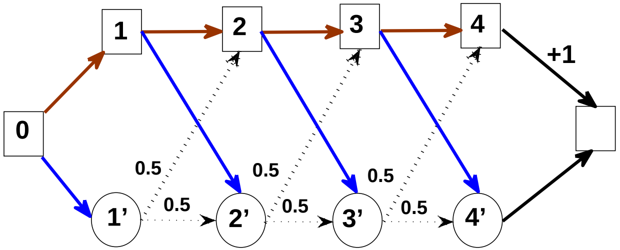
- A stationary policy for the decision maker is a set function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies the state-action π_i that the decision maker will choose when in state i. The MDP is to find a stationary policy to minimize the expected discounted sum over an infinite horizon:

$$\sum_{t=0}^{\infty} \gamma^t c^{\pi_{i^t}}(i^t, i^{t+1}),$$

where $0 \leq \gamma < 1$ is the so-called discount rate.

- Each state (or agent) is myopic and can be selfish. But when every state chooses an optimal action among its available ones, the process reaches optimality and they form an optimal stationary policy.

A Markov Decision Process Example



by Melekopoglou and Condon 1990; actions in red are taken

Applications of The Markov Decision Process

MDP is one of the most fundamental dynamic decision models in

- Mathematical science
- Physical science
- Management science
- Social Science

Modern applications include dynamic planning, reinforcement learning, social networking, and almost all other dynamic/sequential decision making problems.

The LP Form of The Discounted MDP

$$\begin{array}{rccccc} \text{minimize} & \mathbf{c}_1^T\mathbf{x}_1 & \dots & +\mathbf{c}_m^T\mathbf{x}_m & & \\ \text{subject to} & (E_1-\gamma P_1)\mathbf{x}_1 & \dots & +(E_m-\gamma P_m)\mathbf{x}_m & = & \mathbf{e}, \\ & \mathbf{x}_1, & \dots & \mathbf{x}_m, & \geq & \mathbf{0}. \end{array}$$

E_i is the $m \times k_i = |\mathcal{A}_i|$ matrix where the ith row are all ones and everywhere else are zeros, P_i is an $m \times k_i$ column stochastic matrix where each column is the state transition probabilities $P^j(i, i')$, $i = 1, \cdots, m$.

The LP Form of The Discounted MDP

$$\begin{array}{lcccccc} \text{minimize} & \mathbf{c}_1^T\mathbf{x}_1 & \ldots & +\mathbf{c}_m^T\mathbf{x}_m & & \\ \text{subject to} & (E_1-\gamma P_1)\mathbf{x}_1 & \ldots & +(E_m-\gamma P_m)\mathbf{x}_m & = & \mathbf{e}, \\ & \mathbf{x}_1, & \ldots & \mathbf{x}_m, & \geq & \mathbf{0}. \end{array}$$

E_i is the $m \times k_i = |\mathcal{A}_i|$ matrix where the ith row are all ones and everywhere else are zeros, P_i is an $m \times k_i$ column stochastic matrix where each column is the state transition probabilities $P^j(i, i')$, $i = 1, \cdots, m$.

$$\mathbf{e}^T P_i = \mathbf{e}^T \quad \text{and} \quad P_i \geq \mathbf{0}, \quad i = 1, \ldots, m,$$

and $\mathbf{e}$ is the vector of all ones.

The MDP Example in LP form

a:	(0_1)	(0_2)	(1_1)	(1_2)	(2_1)	(2_2)	(3_1)	(3_2)	(4_1)	$(4'_1)$
c:	0	0	0	0	0	0	0	0	1	0
(0)	1	1	0	0	0	0	0	0	0	0
(1)	$-\gamma$	0	1	1	0	0	0	0	0	0
(2)	0	$-\gamma/2$	$-\gamma$	0	1	1	0	0	0	0
(3)	0	$-\gamma/4$	0	$-\gamma/2$	$-\gamma$	0	1	1	0	0
(4)	0	$-\gamma/8$	0	$-\gamma/4$	0	$-\gamma/2$	$-\gamma$	0	$1-\gamma$	0
(4′)	0	$-\gamma/8$	0	$-\gamma/4$	0	$-\gamma/2$	0	$-\gamma$	0	$1-\gamma$

The Discounted MDP Dual Problem

$$
\begin{array}{lrcl}
\text{maximize} & \mathbf{e}^T\mathbf{y} & & \\
\text{subject to} & (E_1 - \gamma P_1)^T\mathbf{y} + \mathbf{s}_1 & = & \mathbf{c}_1, \\
 & \ldots & \ldots & \ldots \\
 & (E_i - \gamma P_i)^T\mathbf{y} + \mathbf{s}_i & = & \mathbf{c}_i, \\
 & \ldots & \ldots & \ldots \\
 & (E_m - \gamma P_m)^T\mathbf{y} + \mathbf{s}_m & = & \mathbf{c}_m, \\
 & (\mathbf{s}_1, \cdots, \mathbf{s}_m) & \geq & \mathbf{0}.
\end{array}
$$

The elements in $\mathbf{s}_i$ are called the slack variables.

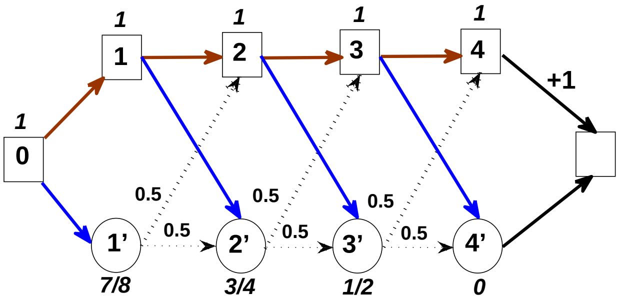
The Interpretations of the Primal and Dual

- Decision $\mathbf{x}_i \in \mathbf{R}^k$ is the state-action frequency for all actions $j \in \mathcal{A}_i$, or the expected present value of the number of times in which an individual is in state i and takes state-action j for all $j \in \mathcal{A}_i$. Thus, solving the discounted MDP primal entails choosing state-action frequencies that minimize the expected present value sum, $\mathbf{c}^T\mathbf{x}$, of total costs.

The Interpretations of the Primal and Dual

- Decision $\mathbf{x}_i \in \mathbf{R}^k$ is the state-action frequency for all actions $j \in \mathcal{A}_i$, or the expected present value of the number of times in which an individual is in state i and takes state-action j for all $j \in \mathcal{A}_i$. Thus, solving the discounted MDP primal entails choosing state-action frequencies that minimize the expected present value sum, $\mathbf{c}^T\mathbf{x}$, of total costs.
- The discounted MDP dual variables $\mathbf{y} \in \mathbf{R}^m$ represent the expected present cost-to-go values of the m states. Solving the dual entails choosing dual variables $\mathbf{y}$, one for each state i, that maximizes $\mathbf{e}^T\mathbf{y}$. It is well known that there exist unique optimal $(\mathbf{y}^*, \mathbf{s}^*)$ where, for each state i, y_i^* is the minimum expected present cost that an individual in state i and its progeny can incur.

Pricing: the Values of the States



Values on each state; actions in red are taken

The Discounted MDP Primal Properties

Lemma

The discounted MDP primal linear programming formulation has the following properties:

1. *The feasible set of the primal is bounded. More precisely, for every feasible* $\mathbf{x} \geq \mathbf{0}$, $\mathbf{e}^T\mathbf{x} = \frac{m}{1-\gamma}$.

The Discounted MDP Primal Properties

Lemma

The discounted MDP primal linear programming formulation has the following properties:

1. *The feasible set of the primal is bounded. More precisely, for every feasible* $\mathbf{x} \geq \mathbf{0}$, $\mathbf{e}^T\mathbf{x} = \frac{m}{1-\gamma}$.
2. *There is a one-to-one correspondence between a (stationary) policy of the original discounted MDP and a basic feasible solution (BFS) of the primal.*

The Discounted MDP Primal Properties

Lemma

The discounted MDP primal linear programming formulation has the following properties:

1. *The feasible set of the primal is bounded. More precisely, for every feasible* $\mathbf{x} \geq \mathbf{0}$, $\mathbf{e}^T\mathbf{x} = \frac{m}{1-\gamma}$.
2. *There is a one-to-one correspondence between a (stationary) policy of the original discounted MDP and a basic feasible solution (BFS) of the primal.*
3. *Every policy or BFS basis has the Leontief substitution form* $A_\pi = I - \gamma P_\pi$.

The Discounted MDP Primal Properties

Lemma

The discounted MDP primal linear programming formulation has the following properties:

1. *The feasible set of the primal is bounded. More precisely, for every feasible* $\mathbf{x} \geq \mathbf{0}$, $\mathbf{e}^T\mathbf{x} = \frac{m}{1-\gamma}$.
2. *There is a one-to-one correspondence between a (stationary) policy of the original discounted MDP and a basic feasible solution (BFS) of the primal.*
3. *Every policy or BFS basis has the Leontief substitution form* $A_\pi = I - \gamma P_\pi$.
4. *Let* $\mathbf{x}^\pi$ *be a basic feasible solution of the primal. Then any basic variable, say* $\mathbf{x}_i^\pi$, *has its value* $1 \leq \mathbf{x}_i^\pi \leq \frac{m}{1-\gamma}$.

Historical Events of the MDP Methods I

- Shapley (1953) and Bellman (1957) developed a method called the value-iteration method to approximate the optimal state values.

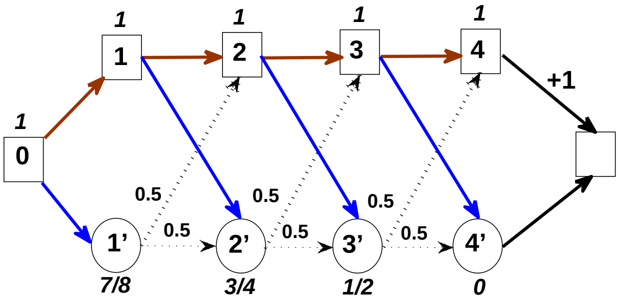
Historical Events of the MDP Methods I

- Shapley (1953) and Bellman (1957) developed a method called the value-iteration method to approximate the optimal state values.
- Another best known method is due to Howard (1960) and is known as the policy-iteration method, which generate an optimal policy in finite number of iterations in a distributed and decentralized way.

Historical Events of the MDP Methods I

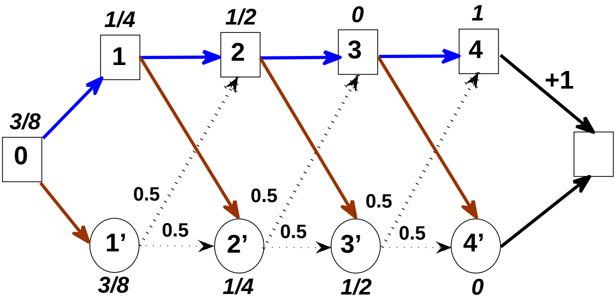
- Shapley (1953) and Bellman (1957) developed a method called the value-iteration method to approximate the optimal state values.
- Another best known method is due to Howard (1960) and is known as the policy-iteration method, which generate an optimal policy in finite number of iterations in a distributed and decentralized way.
- de Ghellinck (1960), D'Epenoux (1960) and Manne (1960) showed that the MDP has an LP representation, so that it can be solved by the simplex method of Dantzig (1947) in finite number of steps, and the Ellipsoid method of Kachiyan (1979) in polynomial time.

Pricing: the Values of the States



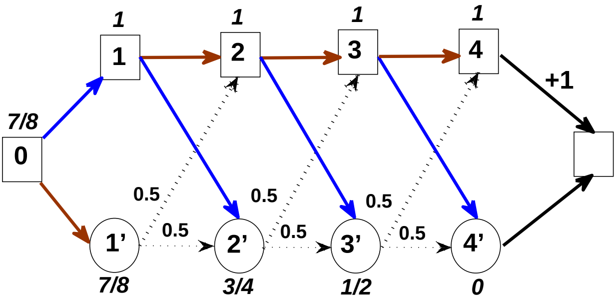
Values on each state; actions in red are taken

The Policy Iteration



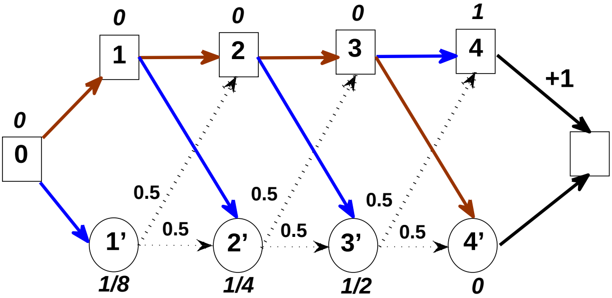
New values on each state; actions in red are taken

The Simplex or Simple Policy Iteration: index rule



New values on each state; actions in red are taken

The Simplex or Simple Policy Iteration: greedy rule



New values on each state; actions in red are taken

Historical Events of the MDP Methods II

- Papadimitriou and Tsitsiclis (1987) gave a theoretical complexity analysis of the MDP and showed that if P_i is deterministic, then the MDP can be solved in strongly polynomial time.

Historical Events of the MDP Methods II

- Papadimitriou and Tsitsiclis (1987) gave a theoretical complexity analysis of the MDP and showed that if P_i is deterministic, then the MDP can be solved in strongly polynomial time.
- Erickson in 1988 showed that successive approximations suffice to produce: (1) an optimal stationary halting policy, or (2) show that no such policy exists in strongly polynomial time algorithm, based on the work of Eaves and Veinott and Rothblum.

Historical Events of the MDP Methods II

- Papadimitriou and Tsitsiclis (1987) gave a theoretical complexity analysis of the MDP and showed that if P_i is deterministic, then the MDP can be solved in strongly polynomial time.
- Erickson in 1988 showed that successive approximations suffice to produce: (1) an optimal stationary halting policy, or (2) show that no such policy exists in strongly polynomial time algorithm, based on the work of Eaves and Veinott and Rothblum.
- Mansour and Singh in 1994 also gave an upper bound on the number of iterations, $\frac{k^m}{m}$, for the policy-iteration method when each state has k actions.

Historical Events of the MDP Methods III

For the discounted MDP:

- Bertsekas in 1987 showed that the value-iteration method converges to the optimal policy in a finite number of iterations.

Historical Events of the MDP Methods III

For the discounted MDP:

- Bertsekas in 1987 showed that the value-iteration method converges to the optimal policy in a finite number of iterations.
- Tseng (1990) showed that the value iteration method generates an optimal policy in polynomial time when the discount γ is fixed.

Historical Events of the MDP Methods III

For the discounted MDP:

- Bertsekas in 1987 showed that the value-iteration method converges to the optimal policy in a finite number of iterations.
- Tseng (1990) showed that the value iteration method generates an optimal policy in polynomial time when the discount γ is fixed.
- Puterman in 1994 showed that the policy-iteration method converges no more slowly than the value iteration method, so that it is also a polynomial-time algorithm.

Historical Events of the MDP Methods III

For the discounted MDP:

- Bertsekas in 1987 showed that the value-iteration method converges to the optimal policy in a finite number of iterations.
- Tseng (1990) showed that the value iteration method generates an optimal policy in polynomial time when the discount γ is fixed.
- Puterman in 1994 showed that the policy-iteration method converges no more slowly than the value iteration method, so that it is also a polynomial-time algorithm.
- Y (2005) showed that the discounted MDP with fixed discount γ can be solved in strongly polynomial time by a combinatorial interior-point method (CIPM).

Polynomial vs Strongly Polynomial

- If the computation time of an algorithm, the total number of basic arithmetic operations needed, of solving the problem with rational data is bounded by a polynomial in m, n, and the total bits, L, of the encoded problem data, then the algorithm is called polynomial-time algorithms.

Polynomial vs Strongly Polynomial

- If the computation time of an algorithm, the total number of basic arithmetic operations needed, of solving the problem with rational data is bounded by a polynomial in m, n, and the total bits, L, of the encoded problem data, then the algorithm is called polynomial-time algorithms.
- The proof of polynomial-time for the value and policy-iteration methods is essentially due to the argument that, when the gap between the objective value of the current policy (or BFS) and the optimal one is small than 2^{-L}, the current policy must be optimal.

Polynomial vs Strongly Polynomial continued

- If the computation time of an algorithm, the total number of basic arithmetic operations needed, of solving the problem is bounded by a polynomial in m and n, then the algorithm is called strongly polynomial-time algorithms.

Polynomial vs Strongly Polynomial continued

- If the computation time of an algorithm, the total number of basic arithmetic operations needed, of solving the problem is bounded by a polynomial in m and n, then the algorithm is called strongly polynomial-time algorithms.
- The proof of a strongly polynomial-time algorithm cannot rely on that gap argument, since the problem data may have irrational entries so that the bit-size of the data can be ∞.

Facts of the Policy Iteration and Simplex Methods

- In practice, the policy-iteration method, including the simple policy-iteration or Simplex method, has been remarkably successful and shown to be most effective and widely used.

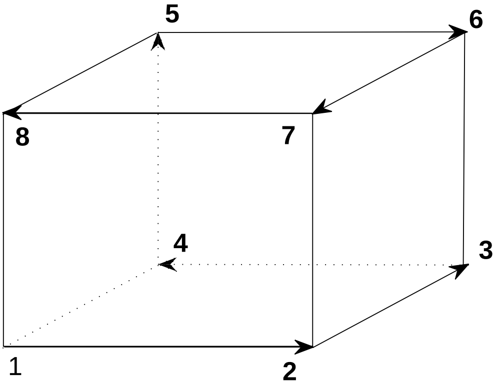
Facts of the Policy Iteration and Simplex Methods

- In practice, the policy-iteration method, including the simple policy-iteration or Simplex method, has been remarkably successful and shown to be most effective and widely used.
- In the past 50 years, many efforts have been made to resolve the worst-case complexity issue of the policy-iteration method or the Simplex method, and to answer the question: is the policy-iteration a strongly polynomial-time algorithm?

Facts of the Policy Iteration and Simplex Methods

- In practice, the policy-iteration method, including the simple policy-iteration or Simplex method, has been remarkably successful and shown to be most effective and widely used.
- In the past 50 years, many efforts have been made to resolve the worst-case complexity issue of the policy-iteration method or the Simplex method, and to answer the question: is the policy-iteration a strongly polynomial-time algorithm?
- In theory, Klee and Minty (1972) have showed that the simplex method, with the greedy (most-negative-reduced-cost) pivoting rule, necessarily takes an exponential number of iterations to solve a carefully designed LP problem.

The Klee and Minty Example I



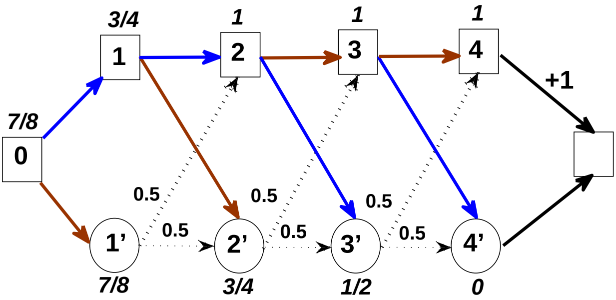
More Negative Results for the Policy-Iteration Method

- A similar negative result of Melekopoglou and Condon (1990) showed that a simple policy-iteration method, where in each iteration only the action for the state with the smallest index is updated, needs an exponential number of iterations to compute an optimal policy for a specific MDP problem regardless of discount rates.

More Negative Results for the Policy-Iteration Method

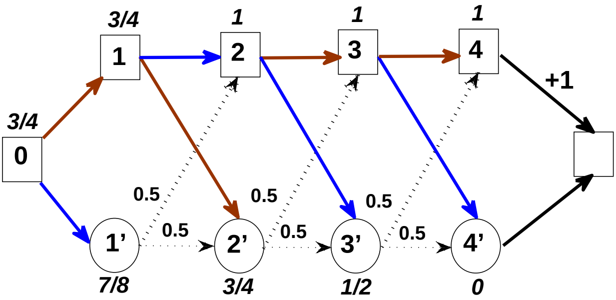
- A similar negative result of Melekopoglou and Condon (1990) showed that a simple policy-iteration method, where in each iteration only the action for the state with the smallest index is updated, needs an exponential number of iterations to compute an optimal policy for a specific MDP problem regardless of discount rates.
- Most recently, Fearnley (2010) showed that the policy-iteration method needs an exponential number of iterations for a undiscounted finite-horizon MDP.

The Simplex or Simple Policy Iteration: index rule II



New values on each state; actions in red are taken

The Simplex or Simple Policy Iteration: index rule III



New values on each state; actions in red are taken

Prior Best Results of the Discount MDP Methods

Value-Iter	Policy-Iter	LP-Alg	Comb IP
$\frac{m^2kL}{1-\gamma}$	$\min\left\{\frac{m^3k^m}{m}, \frac{m^3kL}{1-\gamma}\right\}$	m^3k^2L	$m^4k^4 \cdot \log\frac{m}{1-\gamma}$

where L is a total bits to encode the problem data $(P_i, \mathbf{c}_i, \gamma)$, $i = 1, \ldots, m$, and each state has k actions.

Prior Best Results of the Discount MDP Methods

Value-Iter	Policy-Iter	LP-Alg	Comb IP
$\frac{m^2kL}{1-\gamma}$	$\min\left\{\frac{m^3k^m}{m}, \frac{m^3kL}{1-\gamma}\right\}$	m^3k^2L	$m^4k^4 \cdot \log\frac{m}{1-\gamma}$

where L is a total bits to encode the problem data $(P_i, \mathbf{c}_i, \gamma)$, $i = 1, \ldots, m$, and each state has k actions.

Can we prove the simplex and policy-iteration methods strongly polynomial for the discounted MDP with a fixed rate γ?

Our Result

- The classic simplex method, or the simple policy-iteration method, with the greedy pivoting rule, is a strongly polynomial-time algorithm for MDP with fixed discount rate:

$$\frac{m^2(k-1)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right),$$

and each iteration uses at most $m^2 k$ arithmetic operations.

Our Result

- The classic simplex method, or the simple policy-iteration method, with the greedy pivoting rule, is a strongly polynomial-time algorithm for MDP with fixed discount rate:

$$\frac{m^2(k-1)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right),$$

  and each iteration uses at most m^2k arithmetic operations.

- In general the number of iterations is bounded by $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$, and each iteration uses at most $O(mn)$ arithmetic operations, where n is the total number of actions.

Our Result

- The classic simplex method, or the simple policy-iteration method, with the greedy pivoting rule, is a strongly polynomial-time algorithm for MDP with fixed discount rate:

$$\frac{m^2(k-1)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right),$$

  and each iteration uses at most $m^2 k$ arithmetic operations.
- In general the number of iterations is bounded by $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$, and each iteration uses at most $O(mn)$ arithmetic operations, where n is the total number of actions.
- The policy-iteration method with the all-negative-reduced-cost pivoting rule is at least as good as the simple policy-iteration method, it is also a strongly polynomial-time algorithm with the same iteration complexity bound.

Optimal and Non-Optimal State-Actions

Lemma

There is a unique partition $\mathcal{P} \subseteq \{1, 2, \cdots, n\}$ *and* $\mathcal{O} \subseteq \{1, 2, \cdots, n\}$ *such that for all optimal solution pair* $(\mathbf{x}^*, \mathbf{s}^*)$,

$$x_j^* = 0, \ \forall j \in \mathcal{O}, \quad \text{and} \quad s_j^* = 0, \ \forall j \in \mathcal{P},$$

and there is at least one optimal solution pair $(\mathbf{x}^*, \mathbf{s}^*)$ *that is strictly complementary*,

$$x_j^* > 0, \forall j \in \mathcal{P}, \quad \text{and} \quad s_j^* > 0, \ \forall j \in \mathcal{O},$$

for the DMDP linear program. In particular, every optimal policy $\pi^* \subseteq \mathcal{P}$ *so that* $|\mathcal{P}| \geq m$ *and* $|\mathcal{O}| \leq n - m$.

Optimal and Non-Optimal State-Actions continued

The interpretation of Lemma 2 is as follows: since there may exist multiple optimal policies π^*, $\mathcal{P}$ contains those state-actions each of which appears in at least one optimal policy, and $\mathcal{O}$ contains the rest state-actions neither of which appears in any optimal policy.

Optimal and Non-Optimal State-Actions continued

The interpretation of Lemma 2 is as follows: since there may exist multiple optimal policies π^*, $\mathcal{P}$ contains those state-actions each of which appears in at least one optimal policy, and $\mathcal{O}$ contains the rest state-actions neither of which appears in any optimal policy.

Each state-action in $\mathcal{O}$ is labeled as a non-optimal state-action or simply non-optimal action. Then, any MDP should have no more than $n - m$ non-optimal actions.

High Level Ideas of the Proof

- Create a combinatorial event similar to the one in Vavasis and Y (1994) developed for general LP and Y (2005) for MDP.

High Level Ideas of the Proof

- Create a combinatorial event similar to the one in Vavasis and Y (1994) developed for general LP and Y (2005) for MDP.
- The event will happen in at most a strongly polynomial number of iterations.

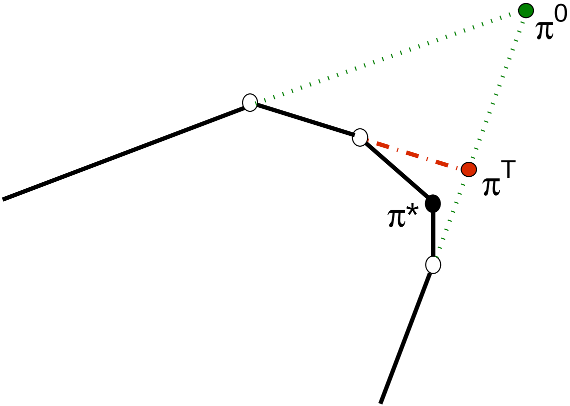
High Level Ideas of the Proof

- Create a combinatorial event similar to the one in Vavasis and Y (1994) developed for general LP and Y (2005) for MDP.
- The event will happen in at most a strongly polynomial number of iterations.
- In particular, after a polynomial number of iterations, a new non-optimal state-action would be eliminated from appearance in any future policies generated by the simplex or policy-iteration method.

High Level Ideas of the Proof

- Create a combinatorial event similar to the one in Vavasis and Y (1994) developed for general LP and Y (2005) for MDP.
- The event will happen in at most a strongly polynomial number of iterations.
- In particular, after a polynomial number of iterations, a new non-optimal state-action would be eliminated from appearance in any future policies generated by the simplex or policy-iteration method.
- The event then repeats for another non-optimal state-action.

Geometric Interpretation in the Dual



The Simplex and Policy-Iteration Methods I

Let π be a policy and ν contain the remaining indexes of the non-basic variables.

$$\begin{array}{rlcl} \text{minimize} & \mathbf{c}_\pi^T \mathbf{x}_\pi \quad + \mathbf{c}_\nu^T \mathbf{x}_\nu & & \\ \text{subject to} & A_\pi \mathbf{x}_\pi \quad + A_\nu \mathbf{x}_\nu & = & \mathbf{e}, \\ & \mathbf{x} = (\mathbf{x}_\pi ; \mathbf{x}_\nu) & \geq & \mathbf{0}, \end{array} \tag{1}$$

The Simplex and Policy-Iteration Methods I

Let π be a policy and ν contain the remaining indexes of the non-basic variables.

$$\begin{array}{lllll} \text{minimize} & \mathbf{c}_\pi^T \mathbf{x}_\pi & +\mathbf{c}_\nu^T \mathbf{x}_\nu & & \\ \text{subject to} & A_\pi \mathbf{x}_\pi & +A_\nu \mathbf{x}_\nu & = & \mathbf{e}, \\ & \mathbf{x} = (\mathbf{x}_\pi; \mathbf{x}_\nu) & & \geq & \mathbf{0}, \end{array} \qquad (1)$$

The (primal) Simplex method rewrites (1) into an equivalent problem

$$\begin{array}{lllll} \text{minimize} & & (\bar{\mathbf{c}}_\nu)^T \mathbf{x}_\nu & & +\mathbf{c}_\pi^T (A_\pi)^{-1} \mathbf{e} \\ \text{subject to} & A_\pi \mathbf{x}_\pi & +A_\nu \mathbf{x}_\nu & = & \mathbf{e}, \\ & \mathbf{x} = (\mathbf{x}_\pi; \mathbf{x}_\nu) & & \geq & \mathbf{0}; \end{array} \qquad (2)$$

where $\bar{\mathbf{c}}$ is called the reduced cost vector:

$$\bar{\mathbf{c}}_\pi = \mathbf{0} \quad \text{and} \quad \bar{\mathbf{c}}_\nu = \mathbf{c}_\nu - A_\nu^T \mathbf{y}^\pi,$$

and

$$\mathbf{y}^\pi = (A_\pi^T)^{-1} \mathbf{c}_\pi.$$

The Simplex Method, Dantzig 1947

- $$0 < \Delta = -\min(\bar{\mathbf{c}}) \quad \text{with} \quad j^{+} = \arg\min(\bar{\mathbf{c}}).$$

The Simplex Method, Dantzig 1947

- $$0 < \Delta = -\min(\bar{\mathbf{c}}) \quad \text{with} \quad j^+ = \arg\min(\bar{\mathbf{c}}).$$
- Let $j^+ \in \mathcal{A}_i$, that is, let j^+ be a state-action controlled by state i, and takes x_{j^+} as the incoming basic variable to replace the old one x_{π_i}.

The Simplex Method, Dantzig 1947

- $$0 < \Delta = -\min(\bar{\mathbf{c}}) \quad \text{with} \quad j^+ = \arg\min(\bar{\mathbf{c}}).$$
- Let $j^+ \in \mathcal{A}_i$, that is, let j^+ be a state-action controlled by state i, and takes x_{j^+} as the incoming basic variable to replace the old one x_{π_i}.
- The method will break a tie arbitrarily, and it updates exactly one state-action in one iteration.

The Simplex Method, Dantzig 1947

- $$0 < \Delta = -\min(\bar{\mathbf{c}}) \quad \text{with} \quad j^+ = \arg\min(\bar{\mathbf{c}}).$$
- Let $j^+ \in \mathcal{A}_i$, that is, let j^+ be a state-action controlled by state i, and takes x_{j^+} as the incoming basic variable to replace the old one x_{π_i}.
- The method will break a tie arbitrarily, and it updates exactly one state-action in one iteration.
- The method repeats with the new policy denoted by π^+ where $\pi_i \in \mathcal{A}_i$ is replaced by $j^+ \in \mathcal{A}_i$.

The Policy-Iteration Method, Howard 1960

- Update every state that has a negative reduced cost. For each state i, let

$$\Delta_i = -\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}) \quad \text{with} \quad j_i^+ = \arg\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}).$$

The Policy-Iteration Method, Howard 1960

- Update every state that has a negative reduced cost. For each state i, let

$$\Delta_i = -\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}) \quad \text{with} \quad j_i^+ = \arg\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}).$$

- Then for every state i such that $\Delta_i > 0$, let $j_i^+ \in \mathcal{A}_i$ replace $\pi_i \in \mathcal{A}_i$ already in the current policy π.

The Policy-Iteration Method, Howard 1960

- Update every state that has a negative reduced cost. For each state i, let

$$\Delta_i = -\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}) \quad \text{with} \quad j_i^+ = \arg\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}).$$

- Then for every state i such that $\Delta_i > 0$, let $j_i^+ \in \mathcal{A}_i$ replace $\pi_i \in \mathcal{A}_i$ already in the current policy π.
- The method repeats with the new policy denoted by π^+.

The Policy-Iteration Method, Howard 1960

- Update every state that has a negative reduced cost. For each state i, let

$$\Delta_i = -\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}) \quad \text{with} \quad j_i^+ = \arg\min_{j \in \mathcal{A}_i}(\bar{\mathbf{c}}).$$

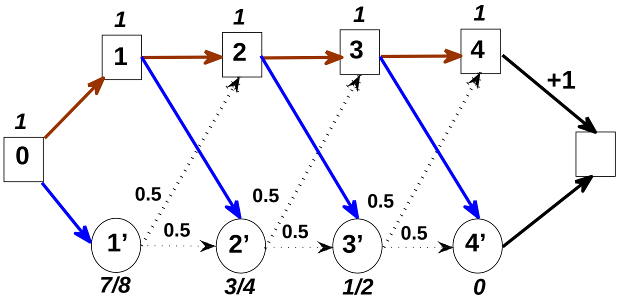
- Then for every state i such that $\Delta_i > 0$, let $j_i^+ \in \mathcal{A}_i$ replace $\pi_i \in \mathcal{A}_i$ already in the current policy π.
- The method repeats with the new policy denoted by π^+.

Therefore, both methods would generate a sequence of polices denoted by $\pi^0, \pi^1, \ldots, \pi^t, \ldots$, starting from any policy π^0.

$$\mathbf{y}^{\pi} = (0;\ 0;\ 0;\ 0;\ -1).$$

a:	(0_1)	(0_2)	(1_1)	(1_2)	(2_1)	(2_2)	(3_1)	(3_2)
c:	0	$-1/8$	0	$-1/4$	0	$-1/2$	0	-1
(0)	1	1	0	0	0	0	0	0
(1)	-1	0	1	1	0	0	0	0
(2)	0	$-1/2$	-1	0	1	1	0	0
(3)	0	$-1/4$	0	$-1/2$	-1	0	1	1
(4)	0	$-1/8$	0	$-1/4$	0	$-1/2$	-1	-1

Reduced Cost at the Current Policy



New values on each state; actions in red are taken

Proof of Strong Polynomiality I

Lemma

Let z^* be the optimal objective value of (1) . Then, in any iteration of the Simplex method from current policy π to new policy π^+

$$z^* \geq \mathbf{c}^T \mathbf{x}^{\pi} - \frac{m}{1-\gamma} \cdot \Delta.$$

Proof of Strong Polynomiality I

Lemma

Let z^* be the optimal objective value of (1) . Then, in any iteration of the Simplex method from current policy π to new policy π^+

$$z^* \geq \mathbf{c}^T \mathbf{x}^{\pi} - \frac{m}{1-\gamma} \cdot \Delta.$$

Moreover,

$$\mathbf{c}^T \mathbf{x}^{\pi^+} - z^* \leq \left(1 - \frac{1-\gamma}{m}\right) \left(\mathbf{c}^T \mathbf{x}^{\pi} - z^*\right).$$

Proof of Strong Polynomiality I

Lemma

Let z^* be the optimal objective value of (1) . Then, in any iteration of the Simplex method from current policy π to new policy π^+

$$z^* \geq \mathbf{c}^T \mathbf{x}^{\pi} - \frac{m}{1-\gamma} \cdot \Delta.$$

Moreover,

$$\mathbf{c}^T \mathbf{x}^{\pi^+} - z^* \leq \left(1 - \frac{1-\gamma}{m}\right) \left(\mathbf{c}^T \mathbf{x}^{\pi} - z^*\right).$$

Therefore, the Simplex method generates a sequence of polices $\pi^0, \pi^1, \ldots, \pi^t, \ldots$ such that

$$\mathbf{c}^T \mathbf{x}^{\pi^t} - z^* \leq \left(1 - \frac{1-\gamma}{m}\right)^t \left(\mathbf{c}^T \mathbf{x}^{\pi^0} - z^*\right).$$

Proof Sketch of the Lemma

The minimal objective value of problem (2) is bounded from below by

$$\mathbf{c}^T \mathbf{x}^\pi - \frac{m}{1-\gamma} \cdot \Delta.$$

Proof Sketch of the Lemma

The minimal objective value of problem (2) is bounded from below by

$$\mathbf{c}^T\mathbf{x}^\pi - \frac{m}{1-\gamma}\cdot\Delta.$$

Problems (1) and (2) share the same objective value.

Proof Sketch of the Lemma

The minimal objective value of problem (2) is bounded from below by

$$\mathbf{c}^T\mathbf{x}^\pi - \frac{m}{1-\gamma}\cdot\Delta.$$

Problems (1) and (2) share the same objective value.

Since at the new policy π^+, the incoming basic variable value is greater than or equal to 1 from Lemma 1, the objective value of the new policy is decreased by at least Δ. Thus,

$$\mathbf{c}^T\mathbf{x}^\pi - \mathbf{c}^T\mathbf{x}^{\pi^+} \geq \Delta \geq \frac{1-\gamma}{m}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right).$$

Proof of Strong Polynomiality II

Lemma

1. *If a policy π is not optimal, then there is a state-action $j \in \pi \cap \mathcal{O}$ (i.e., a non-optimal state-action j in the current policy) such that*

$$s_j^* \geq \frac{1-\gamma}{m^2}\left(\mathbf{c}^T \mathbf{x}^\pi - z^*\right).$$

Proof of Strong Polynomiality II

Lemma

1. *If a policy* π *is not optimal, then there is a state-action* $j \in \pi \cap \mathcal{O}$ *(i.e., a non-optimal state-action* j *in the current policy) such that*

$$s_j^* \geq \frac{1-\gamma}{m^2}\left(\mathbf{c}^T\mathbf{x}^{\pi} - z^*\right).$$

2. *For any sequence of polices* $\pi^0, \pi^1, \ldots, \pi^t, \ldots$ *generated by the Simplex method where* π^0 *is not optimal, let* $j^0 \in \pi^0 \cap \mathcal{O}$ *be the state-action index identified above in the initial policy* π^0*. Then, if* $j^0 \in \pi^t$*, we must have*

$$x_{j^0}^{\pi^t} \leq \frac{m^2}{1-\gamma} \cdot \frac{\mathbf{c}^T\mathbf{x}^{\pi^t} - z^*}{\mathbf{c}^T\mathbf{x}^{\pi^0} - z^*}, \ \forall t \geq 1.$$

Proof Sketch of the Lemma

Since all non-basic variable of $\mathbf{x}^\pi$ have zero values,

$$\mathbf{c}^T\mathbf{x}^\pi - z^* = \mathbf{c}^T\mathbf{x}^\pi - \mathbf{e}^T\mathbf{y}^* = (\mathbf{s}^*)^T\mathbf{x}^\pi = \sum_{j\in\pi} s_j^* x_j^\pi .$$

There must be a state-action $j \in \pi$ such that

$$s_j^* x_j^\pi \geq \frac{1}{m}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right).$$

Proof Sketch of the Lemma

Since all non-basic variable of $\mathbf{x}^\pi$ have zero values,

$$\mathbf{c}^T\mathbf{x}^\pi - z^* = \mathbf{c}^T\mathbf{x}^\pi - \mathbf{e}^T\mathbf{y}^* = (\mathbf{s}^*)^T\mathbf{x}^\pi = \sum_{j\in\pi} s_j^* x_j^\pi .$$

There must be a state-action $j \in \pi$ such that

$$s_j^* x_j^\pi \geq \frac{1}{m}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right).$$

Then,

$$s_j^* \geq \frac{1-\gamma}{m^2}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right) > 0,$$

which also implies $j \in \mathcal{O}$.

Proof Sketch of the Lemma

Since all non-basic variable of $\mathbf{x}^\pi$ have zero values,

$$\mathbf{c}^T\mathbf{x}^\pi - z^* = \mathbf{c}^T\mathbf{x}^\pi - \mathbf{e}^T\mathbf{y}^* = (\mathbf{s}^*)^T\mathbf{x}^\pi = \sum_{j\in\pi} s_j^* x_j^\pi .$$

There must be a state-action $j \in \pi$ such that

$$s_j^* x_j^\pi \geq \frac{1}{m}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right).$$

Then,

$$s_j^* \geq \frac{1-\gamma}{m^2}\left(\mathbf{c}^T\mathbf{x}^\pi - z^*\right) > 0,$$

which also implies $j \in \mathcal{O}$.

Let $j^0 \in \pi^0 \cap \mathcal{O}$ be the index identified at policy π^0. Then, for any policy π^t generated by the Simplex method, if $j^0 \in \pi^t$,

$$\mathbf{c}^T\mathbf{x}^{\pi^t} - z^* = (\mathbf{s}^*)^T\mathbf{x}^{\pi^t} \geq s_{j^0}^* x_{j^0}^{\pi^t} .$$

Proof of Strong Polynomiality III

Theorem

There is a non-optimal state-action in the initial policy π^0 of any policy sequence generated by the Simplex method (with the most-negative-reduced-cost pivoting rule) that would never be in any policy of the sequence after $T := \lceil \frac{m}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right) \rceil$ iterations, that is

$$j^0 \in \pi^0 \cap \mathcal{O}, \quad \textit{but} \quad j^0 \notin \pi^t \; \forall t \geq T+1.$$

Proof Sketch of the Theorem

From Lemma 3, after t iterations of the Simplex method, we have

$$\frac{\mathbf{c}^T\mathbf{x}^{\pi^t} - z^*}{\mathbf{c}^T\mathbf{x}^{\pi^0} - z^*} \leq \left(1 - \frac{1-\gamma}{m}\right)^t.$$

Therefore, after $t \geq T+1$ iterations from the initial policy π^0, $j^0 \in \pi^t$ implies, by Lemma 4,

$$x_{j^0}^{\pi^t} \leq \frac{m^2}{1-\gamma} \cdot \frac{\mathbf{c}^T\mathbf{x}^{\pi^t} - z^*}{\mathbf{c}^T\mathbf{x}^{\pi^0} - z^*} \leq \frac{m^2}{1-\gamma} \cdot \left(1 - \frac{1-\gamma}{m}\right)^t < 1.$$

Proof Sketch of the Theorem

From Lemma 3, after t iterations of the Simplex method, we have

$$\frac{\mathbf{c}^T\mathbf{x}^{\pi^t} - z^*}{\mathbf{c}^T\mathbf{x}^{\pi^0} - z^*} \leq \left(1 - \frac{1-\gamma}{m}\right)^t.$$

Therefore, after $t \geq T+1$ iterations from the initial policy π^0, $j^0 \in \pi^t$ implies, by Lemma 4,

$$x_{j^0}^{\pi^t} \leq \frac{m^2}{1-\gamma} \cdot \frac{\mathbf{c}^T\mathbf{x}^{\pi^t} - z^*}{\mathbf{c}^T\mathbf{x}^{\pi^0} - z^*} \leq \frac{m^2}{1-\gamma} \cdot \left(1 - \frac{1-\gamma}{m}\right)^t < 1.$$

But $x_{j^0}^{\pi^t} < 1$ is a contradiction to Lemma 1, which states that every basic variable value must be greater or equal to 1. Thus, $j^0 \notin \pi^t$ for all $t \geq T+1$.

Proof Sketch of the Main Result

We now repeat the same proof for policy π^{T+1}, if it is not optimal yet, in the policy sequence generated by the Simplex method. Since policy π^{T+1} is not optimal, there must be a non-optimal state-action, $j^1 \in \pi^{T+1} \cap \mathcal{O}$ and $j^1 \neq j^0$ (because of Theorem 5), that would never stay in or return to the policies generated by the Simplex method after $2T$ iterations starting from π^0.

Proof Sketch of the Main Result

We now repeat the same proof for policy π^{T+1}, if it is not optimal yet, in the policy sequence generated by the Simplex method. Since policy π^{T+1} is not optimal, there must be a non-optimal state-action, $j^1 \in \pi^{T+1} \cap \mathcal{O}$ and $j^1 \neq j^0$ (because of Theorem 5), that would never stay in or return to the policies generated by the Simplex method after $2T$ iterations starting from π^0.

In each of these cycles of T Simplex iterations, at least one *new non-optimal state-action* is eliminated from appearance in any of the future policy cycles generated by the Simplex method. However, we have at most $|\mathcal{O}|$ many such non-optimal state-actions to eliminate.

Strong Polynomiality of the Simplex Method

Theorem

The simplex, or simple policy-iteration, method with the most-negative-reduced-cost pivoting rule of Dantzig for solving the discounted Markov decision problem with a fixed discount rate is a strongly polynomial-time algorithm. Starting from any policy, the method terminates in at most $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ *iterations, where each iteration uses* $O(mn)$ *arithmetic operations.*

Strong Polynomiality of the Policy Iteration Method

Corollary

The original policy-iteration method of Howard for solving the discounted Markov decision problem with a fixed discount rate is a strongly polynomial-time algorithm. Starting from any policy, it terminates in at most $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ *iterations.*

Strong Polynomiality of the Policy Iteration Method

Corollary

The original policy-iteration method of Howard for solving the discounted Markov decision problem with a fixed discount rate is a strongly polynomial-time algorithm. Starting from any policy, it terminates in at most $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ *iterations.*

The key is that the state-action with the most-negative-reduced-cost is included in the next policy for the policy-iteration method.

Extensions to other MDPs

Every policy or BFS basis of the undiscounted MDP has the Leontief substitution form:

$$A_\pi = I - P_\pi,$$

where $P_\pi \geq \mathbf{0}$ and the spectral radius of P_π is bounded by $\gamma < 1$. Then, $\mathbf{e}^T(I - P_\pi)^{-1}\mathbf{e} \leq \frac{m}{1-\gamma}$.

Extensions to other MDPs

Every policy or BFS basis of the undiscounted MDP has the Leontief substitution form:

$$A_\pi = I - P_\pi,$$

where $P_\pi \geq \mathbf{0}$ and the spectral radius of P_π is bounded by $\gamma < 1$. Then, $\mathbf{e}^T(I - P_\pi)^{-1}\mathbf{e} \leq \frac{m}{1-\gamma}$.

Corollary

Let every basis of policy π an MDP have the form $I - P_\pi$ where $P_\pi \geq \mathbf{0}$, with a spectral radius less than or equal to a fixed $\gamma < 1$. Then, the Simplex and policy-iteration methods are strongly polynomial-time algorithms. Starting from any policy, each of them terminates in at most $\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ iterations.

Remarks and Open Questions I

- The performance of the simplex method is very sensitive to the pivoting rule.

Remarks and Open Questions I

- The performance of the simplex method is very sensitive to the pivoting rule.
- Tatonnement and decentralized process works under the Markov property.

Remarks and Open Questions I

- The performance of the simplex method is very sensitive to the pivoting rule.
- Tatonnement and decentralized process works under the Markov property.
- Greedy or Steepest Descent works when there is a discount!

Remarks and Open Questions I

- The performance of the simplex method is very sensitive to the pivoting rule.
- Tatonnement and decentralized process works under the Markov property.
- Greedy or Steepest Descent works when there is a discount!
- Multi-updates or pivots work better than a single-update does; policy iteration vs. simplex iteration:

Remarks and Open Questions I

- The performance of the simplex method is very sensitive to the pivoting rule.
- Tatonnement and decentralized process works under the Markov property.
- Greedy or Steepest Descent works when there is a discount!
- Multi-updates or pivots work better than a single-update does; policy iteration vs. simplex iteration: $\frac{n}{1-\gamma} \cdot \log\left(\frac{m}{1-\gamma}\right)$ (only for the policy iteration method) by Hansen, Miltersen, and Zwick (2010).
- The proof techniques are generalized to certain Stochastic Games with fixed discount by Hansen et al. (2010), and certain linear programs by Kitahara and Mizuno (2010).

Remarks and Open Questions II

- What about the value-iteration method?

Remarks and Open Questions II

- What about the value-iteration method?
- Can the iteration bound for the simplex method be reduced to linear in the number of states?

Remarks and Open Questions II

- What about the value-iteration method?
- Can the iteration bound for the simplex method be reduced to linear in the number of states?
- Is the policy iteration method polynomial for the MDP regardless of discount rate γ or input data?

Remarks and Open Questions II

- What about the value-iteration method?
- Can the iteration bound for the simplex method be reduced to linear in the number of states?
- Is the policy iteration method polynomial for the MDP regardless of discount rate γ or input data?
- Is there an MDP algorithm at all whose running time is strongly polynomial regardless of discount rate γ?

Remarks and Open Questions II

- What about the value-iteration method?
- Can the iteration bound for the simplex method be reduced to linear in the number of states?
- Is the policy iteration method polynomial for the MDP regardless of discount rate γ or input data?
- Is there an MDP algorithm at all whose running time is strongly polynomial regardless of discount rate γ?
- Is there a strongly polynomial-time algorithm for LP?