Recent Progress on the Complexity of Solving Markov Decision Processes

Jefferson Huang

Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, USA

January 24, 2014

Overview

- **Markov Decision Processes (MDPs)** provide a framework for modeling and guiding sequential decision making under uncertainty.
- Application areas include Operations Research, Statistics, Economics, Artificial Intelligence, and Finance.
- Recently, there has been renewed interest in the **complexity of algorithms** that solve (i.e. find an optimal policy for) MDPs with finite state and action sets.
- In this talk, we
  - **survey** what is known about the complexity of solution algorithms, and
  - outline directions for **further work**.

Model Definition

- A finite state and action MDP is defined by
  - a set of **states** $\mathbb{X} = \{1, 2, \ldots, n\}$,
  - a set of **actions** $\mathbb{A} = \{1, 2, \ldots, m\}$ and sets of actions $\mathbb{A}(x) \subseteq \mathbb{A}$ available in each state $x \in \mathbb{X}$,
  - **one-step rewards** r, where $r(x, a)$ is the reward earned whenever action $a \in \mathbb{A}(x)$ is performed in state x, and
  - **transition probabilities** p, where $p(y|x, a)$ is the probability that the process transitions to state y given that action $a \in \mathbb{A}(x)$ is performed in state x.
- At each **time step** $t = 0, 1, \ldots$
  - the process is in some state $x_t \in \mathbb{X}$,
  - an action $a_t \in \mathbb{A}(x_t)$ is performed,
  - a reward $r(x_t, a_t)$ is earned, and
  - the state at time $t + 1$ is $y \in \mathbb{X}$ with probability $p(y|x_t, a_t)$.
- For each time step t, a (randomized) **policy** π specifies the probability with which each action $a \in \mathbb{A}(x_t)$ is performed, given the history $x_0 a_0 x_1 a_1 \ldots x_{t-1} a_{t-1} x_t$ of the process up to time t.

Optimization

- Given an MDP, we want to find a policy that is **optimal** over the set of all policies Π^R in some sense.
- Most of the recent complexity results consider the infinite-horizon total **discounted reward** criterion:
  - Each initial state x and policy π defines a stochastic sequence $x_0 a_0 x_1 a_1 \ldots$ with associated expectation operator $\mathbb{E}_x^\pi$.
  - Given a **discount factor** $\beta \in [0, 1)$, the infinite-horizon discounted total reward earned starting from state x under the policy π is

$$v_\beta(x, \pi) \triangleq \mathbb{E}_x^\pi \sum_{t=0}^{\infty} \beta^t r(x_t, a_t).$$

  - A policy π^* is **optimal** under this criterion if

$$v_\beta(x, \pi^*) = \sup_{\pi \in \Pi^R} v_\beta(x, \pi), \quad \text{for all } x \in \mathbb{X}.$$

- Another commonly used criterion is the long-run expected **average reward** per unit time (which we'll consider later).

Finding an Optimal Policy

- A policy ϕ is **stationary** if for each $x \in \mathbb{X}$ it specifies the action to be performed whenever the process is in state x, regardless of how the process got there.
  - The set Π^S of stationary policies can be identified with the set of mappings $\phi : \mathbb{X} \to \mathbb{A}$ satisfying $\phi(x) \in \mathbb{A}(x)$ for all $x \in \mathbb{X}$.
- It is well-known that, if the state & action sets are finite, then there exists a **stationary optimal policy**.
- Two classical algorithms that return a stationary optimal policy after a finite number of iterations are **value iteration** (Shapley 1953, Bellman 1957) and **policy iteration** (Howard 1960).
- **Linear programming** can also be used (Manne 1960, de Ghellinck 1960, d'Epenoux 1963).
  - Policy iteration is equivalent to using the simplex method to solve a certain linear program.

Complexity of Algorithms for MDPs

- An algorithm for solving an MDP is (weakly) **polynomial** if the required number of *arithmetic operations* is bounded above by a polynomial in the number of actions m ($\geq n$) and the bit-size L of the input data.
- If the requisite number of iterations is bounded by a polynomial in m only, the algorithm is **strongly polynomial**.
- We'll now consider both **upper** and **lower** bounds on the number of arithmetic operations required in the worst case for
  - value iteration,
  - policy iteration, and
  - the simplex method.

  After that, we'll consider some recently proposed algorithms that are strongly polynomial under certain conditions.

Value Iteration: Preliminaries

- A **contraction mapping** on a metric space (X, d) is a mapping $A : X \to X$ such that for some $\beta \in [0, 1)$ every $u, v \in X$ satisfies $d(Au, Av) \leq \beta d(u, v)$. Here β is called the *modulus* of the contraction mapping.
- A **fixed point** u of a mapping A satisfies $u = Au$.
- The **Banach fixed-point theorem** states that if (X, d) is complete (i.e. every Cauchy sequence converges), then any contraction mapping A on (X, d) has a unique fixed point u^*, and that for each $u \in X$ and natural number n,

$$d(u^*, A^n u) \leq \frac{\beta^n}{1 - \beta} d(Au, u).$$

  This means that for any $u \in X$, the sequence $\{A^n u\}_{n=0}^{\infty}$ **converges geometrically** to u^*.

Value Iteration: Preliminaries

- Let $B(\mathbb{X})$ be the set of real-valued functions on the state space $\mathbb{X}$, and let the **max-norm** be defined for $u \in B(\mathbb{X})$ by $\|u\|_\infty = \max_{x \in \mathbb{X}} |u(x)|$.
- It's well-known that the mapping $T : B(\mathbb{X}) \to B(\mathbb{X})$ defined for $u \in B(\mathbb{X})$ by

$$Tu(x) = \max_{a \in \mathbb{A}(x)} \{r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a)u(y)\}, \quad x \in \mathbb{X},$$

  is a contraction mapping with modulus β on the complete metric space $(B(\mathbb{X}), \|\cdot\|_\infty)$, implying it has a unique fixed point u^* and that $\{T^n u\}_{n=0}^{\infty}$ converges geometrically to u^*.
- It's also well-known that the **value function**

$$V_\beta(x) = \sup_{\pi \in \Pi^R} v_\beta(x, \pi), \quad x \in \mathbb{X},$$

  is a fixed point of T. Hence $\boxed{u^* = V_\beta.}$

Value Iteration

- For any stationary policy ϕ, let $T_\phi : B(\mathbb{X}) \to B(\mathbb{X})$ be defined for $u \in B(\mathbb{X})$ by

$$T_\phi u(x) = r(x, \phi(x)) + \beta \sum_{y \in \mathbb{X}} p(y|x, \phi(x))u(y), \quad x \in \mathbb{X}.$$

- The **value iteration** algorithm
  1. takes any initial estimate V_0 of the value function at each state x,
  2. iteratively applies T to V_0 (i.e. generates the terms of the sequence $\{T^n V_0\}_{n=0}^{\infty}$) N times, and
  3. given the terminal estimate $V_N \triangleq T^N V_0$, outputs a stationary policy ϕ satisfying $T_\phi V_N = TV_N$.
- The number of iterations N to perform is often determined by a **stopping rule** that gives a lower bound on the performance of ϕ.
- It's well-known that when the state & action sets are finite, then after some finite number of iterations the returned stationary policy ϕ is **optimal**.

Value Iteration: Upper Bound

- Let N^* be the smallest number of iterations needed for value iteration to return an optimal policy.
- Tseng (1990) showed that given *rational input data* with a total bit-size of L,

$$N^* \leq \frac{nL + n\log_2(n)}{1-\beta}.$$

  - This was done by deriving an upper bound for how small $\|V_\beta - V_N\|_\infty$ has to be in order for the returned policy ϕ to be optimal, and using the fact that

$$\|V_\beta - V_N\|_\infty \leq \frac{\beta^N}{1-\beta}\|V_1 - V_0\|_\infty.$$

- This shows that **for a fixed discount factor, value iteration is weakly polynomial**.

Value Iteration: Lower Bounds

- Littman, Dean, & Kaelbling (1995) exhibited a 3-state MDP where
$$N^* \geq \frac{1}{2} \cdot \frac{1}{1-\beta} \log_2 \left(\frac{1}{1-\beta} \right).$$
- Feinberg & Huang (2014) exhibited a similar 3-state MDP where if exact computations are allowed, then N^* may grow arbitrarily quickly with the number of actions.
  - In particular, given the positive integer k, their example has $m = k + 3$ actions. They show that given any increasing sequence $\{M_i\}_{i=1}^{\infty}$ of natural numbers,
$$N^* \geq \frac{M_k}{-\ln(\beta)}.$$
For example, if $M_i = 2^i$ for $i = 1, 2, \ldots,$ then
$$N^* \geq \frac{2^k}{-\ln(\beta)} = \frac{2^m}{-\ln(\beta) \cdot 2^3}.$$

Policy Iteration: Evaluating a Stationary Policy

- Under a stationary policy ϕ, the MDP becomes a **Markov chain** with rewards, where the probability that the process transitions to state y from state x is $p(y|x, \phi(x))$.
- Let I be the $n \times n$ identity matrix, and let P_ϕ be the transition matrix of the Markov chain associated with ϕ.
- Let $v_\phi \in B(\mathbb{X})$ be such that for $x \in \mathbb{X}$, $v_\phi(x) = v_\beta(x, \phi)$.
- Let $r_\phi \in B(\mathbb{X})$ be such that for $x \in \mathbb{X}$, $r_\phi(x) = r(x, \phi(x))$.
- It's well-known that

$$\boxed{v_\phi} = \sum_{t=0}^{\infty} \beta^t P_\phi^t r_\phi = \boxed{(I - \beta P_\phi)^{-1} r_\phi.}$$

- Also, v_ϕ is the fixed point of the contraction mapping T_ϕ.

Policy Iteration: Improving a Stationary Policy

- Let ϕ be a stationary policy.
- Suppose there's a state x^* and a stationary policy ψ such that

$$T_\psi v_\phi(x^*) > v_\phi(x^*).$$

  Then $v_\psi(x^*) > v_\phi(x^*)$.
- Suppose ϕ^* satisfies

$$T_\phi v_{\phi^*}(x) \leq v_{\phi^*}(x), \quad \text{for all } \phi \in \Pi^S,\ x \in \mathbb{X}.$$

  Then $v_\phi(x) \leq v_{\phi^*}(x)$ for all $x \in \mathbb{X}$ and $\phi \in \Pi^S$. Since there is a stationary optimal policy, this means ϕ^* is optimal.

Policy Iteration

- **Policy iteration** (PI) begins with any stationary policy ϕ, and proceeds as follows:
  1. Calculate $v_\phi = (I - \beta P_\phi)^{-1} r_\phi$.
  2. Try to improve ϕ by checking, for each state x, whether there's an action $a \in \mathbb{A}(x)$ satisfying
  $$r(x, a) + \beta \sum_{y \in \mathbb{X}} p(y|x, a) v_\phi(y) > v_\phi(x). \tag{1}$$
  3. If yes,
     3.1 for each $x^* \in \mathbb{X}$ where (1) holds for some action, let $\psi(x^*)$ be any action satisfying (1) when $x = x^*$. For all remaining states x, let $\psi(x) = \phi(x)$.
     3.2 Replace ϕ with ψ and go to step 1.
  4. If no, then ϕ is optimal.
- In step 3.1, we may have a choice as to what action to switch to in a given state x^*.
- For any ϕ and its improvement ψ, $v_\psi > v_\phi$; since $|\Pi^S| \le m^n < \infty$, this means **PI terminates after a finite number of iterations**.

Policy Iteration and Linear Programming

- Let e denote a vector of all ones, and let $[r]_{xa} \triangleq r(x, a)$, $[J]_{xa,y} \triangleq \delta_{xy}$, and $[P]_{xa,y} \triangleq p(y|x, a)$.
- Consider the linear program (LP)

$$\begin{aligned} \max \quad & \rho^T r \\ \text{s.t.} \quad & \rho^T (J - \beta P) = e^T, \ \rho \geq 0. \end{aligned} \qquad (P_\beta)$$

- It's well-known that there's a **1-1 correspondence** between stationary policies and basic feasible solutions to this LP.
- Using the **simplex method** to solve this LP **corresponds to** applying **policy iteration**; note that the reduced "cost" vector for any basis ϕ is

$$\bar{r}_\phi = r - (J - \beta P)(I - \beta P_\phi)^{-1} r_\phi = r + \beta P v_\phi - J v_\phi,$$

and $\bar{r}_\phi(x, a) > 0$ iff. ϕ can be improved by using action a instead of $\phi(x)$ in state x.

PI/Simplex: Pivoting Rules

- Each rule for updating the current policy's selected actions during PI corresponds to a pivoting rule for the simplex method applied to the LP (P_β).
- Two commonly used rules:
  - **Dantzig's** (1947) rule, where the variable with the most positive reduced cost enter the basis.
  - **Howard's** (1960) block pivoting rule, where for each state x^* such that $\bar{r}_\phi(x^*, a) > 0$ for some $a \in \mathbb{A}(x)$, a variable $\rho(x^*, a^*)$ where

$$a^* \in \arg\max_{a \in \mathbb{A}(x^*)} \bar{r}_\phi(x^*, a)$$

    enters the basis. This rule
    - corresponds to updating ϕ to some ψ satisfying $T_\psi v_\phi = T v_\phi$,
    - always pivots the variable Dantzig's rule would've selected into the basis, but
    - might not be justified for general LPs.

PI/Simplex: Upper Bounds (Discount Factor Dependent)

- Let N^* denote the number of iterations PI/simplex needs to return an optimal policy.
- Note that the number of arithmetic operations required for each iteration of PI/simplex is at most proportional to nm (single pivot per iteration) or $n^2 m$ (Howard's rule).
- Meister & Holzbaur (1986) showed that under **Howard's** rule,

$$N^* \leq C \cdot \frac{nL}{-\log(\beta)}$$

for some constant C, and hence that **for a fixed discount factor**, PI/simplex with Howard's rule is **weakly polynomial**.

PI/Simplex: Upper Bounds (Discount Factor Dependent)

- Ye (2011) showed that under both **Dantzig's** and **Howard's** rule,

$$N^* \leq (m-n)\left\lceil \frac{n}{1-\beta} \ln\left(\frac{n^2}{1-\beta}\right)\right\rceil .$$

- Hansen, Miltersen, and Zwick (2013) improved Ye's bound for **Howard's** rule by a factor of n:

$$N^* \leq (m-n)\left\lceil \frac{1}{1-\beta} \ln\left(\frac{n}{1-\beta}\right)\right\rceil ,$$

  and extended it to strategy iteration for *2-player turn-based stochastic games*.

- Scherrer (2013) got rid of the $\ln(n)$ term in the bound for **Howard's** rule:

$$N^* \leq (m-n)\left\lceil \frac{1}{1-\beta} \ln\left(\frac{1}{1-\beta}\right)\right\rceil .$$

PI/Simplex: Upper Bounds (Discount Factor Dependent)

- Scherrer (2013) also showed that under **Dantzig's** rule,

$$N^* \leq (m-n) \cdot n \left[\frac{2}{1-\beta} \ln \left(\frac{1}{1-\beta} \right) \right].$$

- In summary, under **Howard's rule**

$$\boxed{N^* = O\left(\frac{m}{1-\beta} \log \left(\frac{1}{1-\beta} \right) \right),}$$

while under **Dantzig's rule**

$$\boxed{N^* = O\left(\frac{nm}{1-\beta} \log \left(\frac{1}{1-\beta} \right) \right).}$$

PI/Simplex: Upper Bounds (Strongly Polynomial)

- Post & Ye (2013) showed that, if all transitions in the MDP are **deterministic**, then under **Dantzig's** rule

$$N^* \leq C \cdot n^3 m^2 \log^2(n)$$

  for some constant C.
  - Hansen, Kaplan, and Zwick (2014) improved this bound by a factor of n.
- Even & Zadorojniy (2012) showed that for MDPs satisfying a *coupling property* (e.g. controlled discrete-time **M/M/1 queues**), then under the **Gass-Saaty** (1955) shadow vertex pivoting rule

$$N^* \leq m.$$

  - PI/simplex with the Gass-Saaty rule is equivalent to an algorithm proposed by Zadorojniy, Even, and Schwartz (2009).

PI/Simplex: Upper Bounds (Indep. of both β and L)

- Mansour & Singh (1999) showed that if $\overline{m} \triangleq \max_{x \in \mathbb{X}} |\mathbb{A}(x)|$, then under **Howard's** rule

$$N^* \leq C \cdot \frac{\overline{m}^n}{n}$$

for some constant C.

- This is still the best known general upper bound for Howard's rule that's independent of both the discount factor β and the bit-size L of the data.

PI/Simplex: Upper Bounds (Summary)

- PI/simplex is **strongly polynomial** in the following cases:
  - under both Howard's and Dantzig's rule for a **fixed discount factor**, with complexity

    $$O(n^2 m \cdot m) = O(n^2 m^2) \quad \text{and} \quad O((n^2 + nm) \cdot nm) = O(n^2 m^2),$$

    respectively;
  - under Dantzig's rule for **deterministic** MDPs, with complexity

    $$O((n^2 + nm) \cdot n^2 m^2 \log^2(n)) = O(n^3 m^3 \log^2(n));$$

  - under the Gass-Saaty rule for **controlled random walks**, with complexity

    $$O((n^2 + nm) \cdot m) = O(nm^2).$$

PI/Simplex: Polynomial Lower Bounds

- Andersson, Hansen, & Miltersen (2009) exhibited an MDP with 2 actions per state where under **Howard's** rule and for **any discount factor**,

$$N^* \geq C \cdot n$$

for some constant C.

PI/Simplex: Exponential Lower Bounds

- Melekopoglou & Condon (1994) exhibited an MDP where, under **Bland's** (1977) anticycling rule,

  $$N^* \geq C \cdot 2^n$$

  for some constant C.
- Hollanders, Delvenne, & Jungers (2012) modified an example of Fearnley (2010) to show that for a **suitably large discount factor**, under **Howard's** rule

  $$N^* \geq C \cdot 2^n$$

  for some constant C.

PI/Simplex: Subexponential Lower Bounds

- Friedmann (2011) exhibited an MDP where, for a suitably large discount factor, under **Zadeh's** (1980) least-entered rule

$$N^* \geq 2^{C \cdot \sqrt{n}}$$

  for some constant C.
- Friedmann (2012) exhibited an MDP where, for a suitably large discount factor, under **Cunningham's** (1979) round-robin rule

$$N^* \geq 2^{C \cdot \sqrt{n}}$$

  for some constant C.

PI/Simplex: Subexponential Lower Bounds

- Friedmann, Hansen, and Zwick (2011) exhibited an MDP where, for a certain discount factor, under Dantzig's (1963) **random-edge** rule the expected number of iterations needed is

$$2^{C \cdot \sqrt[4]{n}}$$

  for some constant C.
- They also exhibited an MDP where, for a certain discount factor, under Matoušek, Sharir, & Welzl's (1996) **random-facet** rule the expected number of iterations needed is

$$2^{C \cdot \sqrt{n} / \log^{c}(n)}$$

  for some constant C.

PI/Simplex: Lower Bounds (Summary)

- PI/simplex can be **exponential** in the following cases:
  - under **Bland's** rule;
  - under **Howard's** rule, for a large enough discount factor.
- PI/simplex can be **subexponential** under the following *history-dependent* pivoting rules:
  - under **Zadeh's** rule, for a large enough discount factor;
  - under **Cunningham's** rule, for a large enough discount factor.
- PI/simplex can require an **expected subexponential** number of arithmetic operations under the following *randomized* pivoting rules:
  - Dantzig's **random-edge** rule, for some discount factor;
  - Matoušek, Sharir, & Welzl's **random-facet** rule, for some discount factor.

New Strongly Polynomial Algorithms

- Before his 2011 result on PI, Ye (2005) presented an interior point algorithm requiring

$$O\left(m^4 \log\left(\frac{m}{1-\beta}\right)\right)$$

arithmetic operations to return an optimal policy.
  - This was the **first** algorithm shown to be strongly polynomial for MDPs with a fixed discount factor.
- Zadorojniy, Even, and Schwartz (2009) gave a strongly polynomial algorithm for **controlled random walks**, which Even & Zadorojniy (2012) showed to be equivalent to simplex with the Gass-Saaty rule. It requires

$$O((n^2 + nm) \cdot m) = O(nm^2)$$

arithmetic operations.

New Strongly Polynomial Algorithms

- Andersson & Vorobyov (2006) proposed a strongly polynomial algorithm that solves **deterministic** discounted MDPs using

$$O(n^2 m)$$

  arithmetic operations.
- Madani, Thorup, & Zwick (2010) gave two new strongly polynomial algorithms for **deterministic** discounted MDPs; one requires

$$O(nm + n^2 \log(n))$$

  arithmetic operations, and the other requires

$$\Theta(nm)$$

  arithmetic operations.

Future Directions

1. Consider the complexity of algorithms for **average-reward** MDPs.
2. Exhibit LPs/MDPs on which the simplex method is **not strongly polynomial**.
3. Develop **sufficient conditions** for the simplex method to be strongly polynomial.

Average-Reward MDPs

- The long-run expected **average reward** per unit time earned under the policy $\pi \in \Pi^R$ starting from state $x \in \mathbb{X}$ is

$$g(x, \pi) \triangleq \liminf_{N \to \infty} \mathbb{E}_x^\pi \frac{1}{N} \sum_{t=0}^{N-1} r(x_t, a_t).$$

- A policy π^* is **optimal** under the average-reward criterion if $g(x, \pi^*) = \sup_{\pi \in \Pi^R} g(x, \pi)$ for all $x \in \mathbb{X}$.
- Similarly to the discounted case,
  - stationary optimal policies exist when the state & action sets are finite, and
  - value iteration, policy iteration, and linear programming methods exist.

Average-Reward MDPs

- If the MDP is **deterministic**, then the average-reward problem reduces to the classical problem of finding a **minimum mean weight cycle** in a directed graph, which is solvable in strongly polynomial time (e.g. Karp 1978; Young, Tarjan, & Orlin 1991).
- For the stochastic average-reward case, there are **relatively few strong polynomiality results**.
  - The algorithm of Zadorojniy, Even, and Schwartz (2009) also solves average-reward controlled random walks using $O(nm^2)$ arithmetic operations.
  - Feinberg & Huang (2013) showed that policy iteration is strongly polynomial for MDPs modeling replacement & mainteneance problems with a fixed failure probability.
  - Akian & Gaubert (2013) showed that if there's a state that's recurrent under all stationary policies, then policy iteration is strongly polynomial.

Examples Where Simplex Isn't Strongly Polynomial

- We conjecture that there is a **unichain** MDP, i.e. where the Markov chain induced by every stationary policy has a single recurrent class, on which PI/simplex for average rewards will perform badly (e.g. be exponential).
- There may also be an MDP with a **majorant**, i.e. where there exists a number $q(x)$ for each $x \in \mathbb{X}$ satisfying

$$q(y) \geq p(y|x,a) \; \forall \; x, y \in \mathbb{X} \; \& \; a \in \mathbb{A}(x) \quad \text{and} \quad \sum_{y \in \mathbb{X}} q(y) < 2,$$

  on which PI/simplex for average rewards does badly.
- An MDP with a majorant can be reduced to a discounted MDP with a **negative discount factor**. We conjecture that PI/simplex for discounted rewards may not be strongly polynomial for such MDPs either.

Conditions Ensuring Simplex is Strongly Polynomial

- Kitahara & Mizuno (2011) used Ye's (2011) analysis to show that if an LP with n constraints and m variables has an optimal solution, and the values of all the positive elements of any basic feasible solution are between δ and γ, then under both **Dantzig's** rule and the **best-improvement** rule, the simplex method will generate at most

$$m \left\lceil n \cdot \frac{\gamma}{\delta} \ln \left(n \cdot \frac{\gamma}{\delta} \right) \right\rceil$$

  distinct basic feasible solutions.
  - For the LP (P_β), $\delta = 1$ and $\gamma = n/(1-\beta)$.
- Are there other conditions that imply the simplex method is strongly polynomial?