Recent Progress on the Complexity of Solving Markov Decision Processes

Jefferson Huang

1 Introduction

The complexity of algorithms for solving Markov Decision Processes (MDPs) with finite state and action spaces has seen renewed interest in recent years. New strongly polynomial bounds have been obtained for some classical algorithms, while others have been shown to have worst case exponential complexity. In addition, new strongly polynomial algorithms have been developed. We survey these results, and identify directions for further work.

In the following subsections, we define the model, the two optimality criteria we consider (discounted and average rewards), the classical value iteration, policy iteration algorithms, and how to find an optimal policy via linear programming. In Section 2, we review the literature on the complexity of algorithms for sovling discounted and average-reward problems. Finally, in Section 3 we consider some directions for further work.

1.1 Model Definition

Let $\mathbb{X} = \{1, 2, \dots, n\}$ denote the state space, and let $\mathbb{A} = \{1, 2, \dots, m\}$ denote the action space. For $x \in \mathbb{X}$, let $\mathbb{A}(x)$ be the nonempty set of actions available at state x. At each time step $t \in \{0, 1, \dots\}$, the process is in some state $x \in \mathbb{X}$. If the action $a \in \mathbb{A}(x)$ is performed, a one-step reward $r(x, a)$ is earned and the process transitions to state $y \in \mathbb{X}$ with probability $p(y|x, a)$. Given an initial state $x_0 \in \mathbb{X}$, at time $t > 0$ a particular *history* $h_t = x_0 a_0 x_1 a_1 \dots a_{t-1} x_t$ of the process will have been realized, where x_k and $a_k \in \mathbb{A}(x_k)$ is the state and action taken at time k, respectively. A *trajectory* is a sequence $x_0 a_0 x_1 a_1 \dots$.

1.2 Optimality Criteria

A *policy* is any rule that prescribes how actions should be selected under any realization of the process. For example, a policy may stipulate that given the history h_t of the process, the action $a \in \mathbb{A}(x_t)$ should be performed with probability $\pi(a|h_t)$; such a policy is a *randomized policy*, and is the most general kind of policy considered here. The set of all such policies is denoted by Π^R. Of particular note for finite state and action MDPs is the set Π^S of *stationary policies*, where $\phi \in \Pi^S$ is a mapping from $\mathbb{X}$ into $\mathbb{A}$ such that $\phi(x) \in \mathbb{A}(x)$ for each $x \in \mathbb{X}$; under ϕ, the action $\phi(x)$ is performed whenever the process is in state x. Note that Π^S can be viewed as a subset of Π^R.

To evaluate a given policy $\pi \in \Pi^R$, a *criterion* f is selected, which assigns a number $f(x,\pi)$ to each initial state $x \in \mathbb{X}$. The policy π^* is *optimal at state* x if $f(x,\pi^*) = \sup_{\pi\in\Pi^R} f(x,\pi)$; an *optimal policy* is optimal at every initial state.

Two commonly used criteria are the infinite-horizon total discounted reward and the long-run expected average reward per unit time. In particular, let $\mathbb{E}_x^\pi$ denote the expectation operator associated with the probability distribution on the set of possible trajectories of the process with initial state x. Then, given the discount factor $\beta \in [0,1)$, the *infinite-horizon total discounted reward* earned by starting in state x and following the policy π is

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

and the corresponding *long-run average reward per unit time*, also called the *gain*, is

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

It is well-known that, if the state and action spaces are finite, then there is a stationary optimal policy under both the discounted and average-reward criteria; see e.g. Puterman [22, pg. 154, 451]. In particular, the *value function* $V_\beta(x) \triangleq \sup_{\pi\in\Pi^R} v_\beta(x,\pi)$ and $v_\beta(x,\phi)$ for any $\phi \in \Pi^S$ uniquely satisfy

$$V_\beta(x) = \max_{a\in\mathbb{A}(x)} \{r(x,a) + \beta \sum_{y\in\mathbb{X}} p(y|x,a) V_\beta(y)\} \triangleq TV_\beta(x), \quad x \in \mathbb{X}, \tag{1}$$

and

$$v_\beta(x,\phi) = r(x,\phi(x)) + \beta \sum_{y\in\mathbb{X}} p(y|x,a) v_\beta(y,\phi)\} \triangleq T_\phi v_\beta(x,\phi), \quad x \in \mathbb{X}, \tag{2}$$

respectively. Hence any stationary policy ϕ^* satisfying $T_{\phi^*}V_\beta = TV_\beta$ is such that $v_\beta(x,\phi^*) = V_\beta(x)$ for all $x \in \mathbb{X}$, i.e. is optimal. Equation (1) is called the *optimality equation* for the discounted-reward criterion.

For the average-reward criterion, there exists a stationary policy ϕ^* such that, for some real-valued functions g^* and h^* on the state space,

$$g^*(x) = \sum_{y\in\mathbb{X}} p(y|x,\phi^*(x)) g^*(y) = \max_{a\in\mathbb{A}(x)} \{\sum_{y\in\mathbb{X}} p(y|x,a) g^*(y)\}, \quad x \in \mathbb{X}, \tag{3}$$

and

$$\begin{aligned} g^*(x) + h^*(x) &= r(x,\phi^*(x)) + \sum_{y\in\mathbb{X}} p(y|x,\phi^*(x)) h^*(y) \\ &= \max_{a\in\mathbb{A}(x)} \{r(x,a) + \sum_{y\in\mathbb{X}} p(y|x,a) h^*(y)\}, \quad x \in \mathbb{X}; \end{aligned} \tag{4}$$

further, any such ϕ^* is optimal. In particular, $g^*(x) = \sup_{\pi\in\Pi^R} g(x,\pi)$ for each $x \in \mathbb{X}$. Equations (3) and (4) are called the *canonical equations*.

1.3 Solution Methods

Three classical methods for finding an optimal policy are *value iteration*, *policy iteration*, and solving an associated *linear programming* problem.

1.3.1 Value Iteration: Discounted Rewards

Recall from (1) that for any real-valued function u on $\mathbb{X}$, the operator T is given by

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

It is well-known that T is a contraction mapping with modulus β on the space of real-valued functions on $\mathbb{X}$ with respect to the *max-norm* defined for $u : \mathbb{X} \to \mathbb{R}$ by $\|u\| = \max_{x \in \mathbb{X}} |u(x)|$. In other words, for any real-valued functions u, u' on $\mathbb{X}$, $\|Tu - Tu'\| \leq \beta \|u - u'\|$. This can be proved by applying T to the inequalites $u(x) \leq u'(x) + \|u - u'\|$ and $u'(x) \leq u(x) + \|u - u'\|$, $x \in \mathbb{X}$, and noting that $T(u + c) = Tu + \beta c$ for any constant c and $Tu \geq Tv$ if $u \geq v$.

Since the space of real-valued functions on $\mathbb{X}$ can be identified with $\mathbb{R}^n$, which is complete under the max-norm, the Banach fixed-point theorem implies that T has a unique fixed point V^*, i.e. $V^* : \mathbb{X} \to \mathbb{R}$ uniquely satisfies $V^* = TV^*$. Further, given any function $u : \mathbb{X} \to \mathbb{R}$,

$$\|V^* - T^n u\| \leq \frac{\beta^n}{1 - \beta} \|Tu - u\|, \quad n = 1, 2, \ldots, \tag{5}$$

i.e. the sequence $\{T^n u\}_{n \geq 0}$ converges geometrically in max-norm to V^*. Since V_β satisfies $V_\beta = TV_\beta$, we have $V^* = V_\beta$, and (5) immediately suggests a means of obtaining an arbitrarily good approximation of V_β and a corresponding policy:

Algorithm 1 Value Iteration

1: Select a function $u : \mathbb{X} \to \mathbb{R}$, and select a stopping condition.
2: **repeat**
3: Let $u = Tu$.
4: **until** The stopping condition is met.
5: Let ψ be any stationary policy satisfying $T_\psi u = Tu$.

There are a number of stopping conditions that provide a lower bound on the performance of the policy ψ obtained via value iteration; see e.g. [22, §6.3 & §6.6]. In fact, one can show that after a finite number of iterations, ψ must be optimal. In particular, letting $v_\phi(x) \triangleq v_\beta(x, \phi)$ for $\phi \in \Pi^S$ and $x \in \mathbb{X}$, this can be done by first verifying that if value iteration is terminated after N iterations, then

$$\|V_\beta - v_\psi\| \leq \frac{2\beta^N}{(1 - \beta)^2} \|Tu - u\|.$$

The proof is completed by noting that, since $\mathbb{X}$ and $\mathbb{A}$ are finite, the set F^- of nonoptimal stationary policies is finite; hence after any number N of iterations satisfying

$$\frac{2\beta^N}{(1-\beta)^2}\|Tu-u\| < \min_{\phi\in F^-}\|V_\beta - v_\phi\|,$$

the returned policy ψ is optimal. The fact that $\beta \in [0,1)$ ensures that the requisite number of iterations is finite.

1.3.2 Value Iteration: Average Rewards

We only briefly describe value iteration under the average reward criterion, since it is not considered in any of the results described in the sequel. Here value iteration is as in Algorithm 1, except that the operator T is replaced with U, defined for $u : \mathbb{X} \to \mathbb{R}$ as

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

Under certain conditions, the difference $U^{n+1}u(x) - U^n u(x)$ converges to the optimal gain $g^*(x)$ for each $x \in \mathbb{X}$ and stopping conditions exist that provide a lower bound on the performance of ψ. One such condition is that the MDP is both *unichain* and aperiodic, i.e. every stationary policy induces an aperiodic Markov chain with a single recurrent class; see e.g. [22, §8.5, §9.4, & §9.5.3].

1.3.3 Policy Iteration: Discounted Rewards

Recall that for any stationary policy ϕ, the function $v_\beta(x,\phi)$ uniquely satisfies

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

One way to show this is to consider the system of equations $u = T_\phi u$. Letting $r_\phi(x) \triangleq r(x,\phi(x))$ and letting P_ϕ denote the transition matrix of the Markov chain induced by ϕ, we can rewrite $u = T_\phi u$ as $(I-\beta P_\phi)u = r_\phi$, where I is the $n \times n$ identity matrix. Since $\beta^t P_\phi^t$ tends to a zero matrix as $t \to \infty$, the matrix $I - \beta P_\phi$ is invertible, and $(I-\beta P_\phi)^{-1} = \sum_{t=0}^\infty \beta^t P_\phi^t$; this is because

$$(I-\beta P_\phi)\sum_{t=0}^{N-1}\beta^t P_\phi^t = I - \beta^N P_\phi^N$$

tends to I as $N \to \infty$, which means that the determinant of $I - \beta P_\phi$ cannot be zero. Hence $u = (I-\beta P_\phi)^{-1} r_\phi = \sum_{t=0}^\infty \beta^t P_\phi^t r_\phi = v_\phi$, and so v_ϕ uniquely satisfies $v_\phi = T_\phi v_\phi$.

Another way is to first show that T_ϕ is a contraction mapping on the space of real-valued functions on $\mathbb{X}$, and then to show that v_ϕ satisfies $v_\phi = T_\phi v_\phi$ by conditioning on the first transition under ϕ. Invoking the Banach fixed-point

theorem in turn ensures the uniqueness of v_ϕ, and also implies that for any $u : \mathbb{X} \to \mathbb{R}$ the sequence $\{T_\phi^n u\}_{n=0}^\infty$ converges geometrically in max-norm to v_ϕ.

A benefit of the contraction mapping approach is that it can be used to justify a means of deciding whether a given stationary policy can be improved, which forms the basis of the policy iteration algorithm to be described below. In particular, suppose the stationary policy ϕ is such that

$$T_\psi v_\phi(x^*) > v_\phi(x^*), \quad \text{for some } \psi \in \Pi^S \text{ and } x^* \in \mathbb{X}. \tag{6}$$

Noting that since T_ϕ is a monotone operator, $T_\psi^k v_\phi(x^*) > v_\phi(x^*)$ for some $k \in \mathbb{N}$ implies $T_\psi^{k+1} v_\phi(x^*) \geq T_\psi v_\phi(x^*) > v_\phi(x^*)$, we have by induction that $T_\psi^n v_\phi(x^*) > v_\phi(x^*)$ for each $n \in \mathbb{N}$. Since $\lim_{n\to\infty} T_\psi^n v_\phi(x^*) = v_\psi(x^*)$ by the Banach fixed point theorem, this means that $v_\psi(x^*) > v_\phi(x^*)$, i.e. using ψ from the initial state x^* is strictly better than using ϕ.

On the other hand, suppose that ϕ^* is such that

$$T_\phi v_{\phi^*}(x) \leq v_{\phi^*}(x), \quad \text{for all } \phi \in \Pi^S \text{ and } x \in \mathbb{X}, \tag{7}$$

i.e. $T v_{\phi^*} = v_{\phi^*}$. Then since V_β is the unique fixed point of T, ϕ^* is optimal.

Conditions (6) and (7) suggest that, to find an optimal stationary policy, one can start with any $\phi \in \Pi^S$ and iteratively try to improve it by checking if (6) holds; if not, condition (7) holds, implying that the current policy is optimal. This process is referred to as the policy iteration algorithm:

Algorithm 2 Policy Iteration

1: Select any $\phi \in \Pi^S$.
2: **while** $\|T v_\phi - v_\phi\| > 0$ **do**
3: Let ψ be any policy satisfying $T_\psi v_\phi > v_\phi$.
4: Set $\phi = \psi$.

The monotonicity of T implies that the condition in line 2 above holds iff. $T v_\phi(x^*) > v_\phi(x^*)$ for some $x^* \in \mathbb{X}$, which holds iff. condition (6) holds. This is because if $T v_\phi(x^*) < v_\phi(x^*)$, then $T^n v_\phi(x^*) < v_\phi(x^*)$ would hold for all $n \in \mathbb{N}$, implying that $\sup_{\pi \in \Pi^R} v(x^*, \pi) = V_\beta(x^*) < v_\phi(x^*)$, a contradiction. Also, note that in line 3 we may have a choice as to which policy ψ to use; as we will see in Section 1.3.5 below, using the policy iteration algorithm with a certain rule for selecting the improved policy ψ is equivalent to using the simplex method with a certain pivoting rule to solve a certain linear program.

Like the value iteration algorithm (Algorithm 1), the policy iteration algorithm is guaranteed to produce an optimal policy in a finite number of iterations. This can be seen by recalling that if the current policy ϕ is updated to ψ, then $v_\psi > v_\phi$ and hence $\phi \neq \psi$; this implies that the number of policy updates can be at most the number of stationary policies, which is finite because the number of states and actions is finite.

Further, if each updated policy ψ in line 3 of Algorithm 2 is selected so that $T_\psi v_\phi = T v_\phi$, in which case the algorithm becomes the classical *Howard's*

policy iteration algorithm [11, p. 84], then the rate of convergence to an optimal policy is geometric. In particular, letting ϕ^k be the k^{th} policy produced by the algorithm, we have $T_{\phi^{k+1}} v_{\phi^k} = T v_{\phi^k} \geq v_{\phi^k}$, which by induction implies that $T^n_{\phi^{k+1}} v_{\phi^k} \geq T v_{\phi^k}$ for every $n \in \mathbb{N}$. Letting $n \to \infty$, this means $v_{\phi^{k+1}} \geq T v_{\phi^k}$ on each iteration k; by induction, this implies that the k^{th} policy produced by the algorithm satisfies $v_{\phi^k} \geq T^k v_{\phi^0}$, where ϕ^0 is the initial policy. Hence $\|V_\beta - v_{\phi^k}\| \leq \|V_\beta - T^k v_{\phi^0}\| \leq \beta^k \|V_\beta - v_{\phi^0}\|$. This property of Howard's policy iteration plays a key role in Meister & Holzbaur's [18] proof that Howard's policy iteration takes (weakly) polynomial time. It was also used in Hansen et al.'s [9] and Scherrer's [23] recent approaches to improving Ye's [27] proof that, given a fixed discount factor, Howard's policy iteration algorithm runs in strongly polynomial time. More will be said about this in Section 2.3.1 below.

Finally, we remark that Howard's policy iteration algorithm is also equivalent to using Newton's method to solve the functional equation $Tu - u = 0$. This representation can be used to prove the convergence of Howard's policy iteration algorithm in more general settings; see e.g. Puterman [22, §6.4.3-6.4.4].

1.3.4 Policy Iteration: Average Rewards

As was alluded to in the description of value iteration for average-reward MDPs in Section 1.3.2, the structure of the Markov chains induced by the stationary policies for the MDP plays a more prominent role here than in discounted-reward problems. Under certain conditions, the analysis of and algorithms for solving average-reward MDPs can be simplified, while in others a more elaborate analysis is needed. Except for the results of Zadorojniy et al. [28] and Even & Zadorojniy [3], which pertain to controlled random walks, and Melekopoglou & Condon [19], which involves an example under which the optimal policy is optimal under both criteria, all of the results described in the sequel apply to MDPs that are *unichain*, i.e. the Markov chain associated with every stationary policy consists of a single recurrent class and a possibly empty set of transient states. We can also assume without loss of generality that the Markov chain induced by every stationary policy is aperiodic, since an MDP with periodic stationary policies can be transformed in a way that makes all stationary policies aperiodic and perserves the set of optimal policies; see e.g. Puterman [22, §8.5.4].

Under the unichain condition, the gain under every stationary policy is constant. This is because, under this condition, the Cesàro limit

$$\lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} P^t = P^*$$

of any $n \times n$ transition matrix P has identical rows, each of which gives the stationary distribution of P. Hence for any stationary policy ϕ, letting $g_\phi(x) = g(x, \phi)$ and $r_\phi(x) = r(x, \phi(x))$ for $x \in \mathbb{X}$ and letting P_ϕ denote the transition

matrix associated with ϕ and P_ϕ^* its corresponding Cesàro limit, we have that

$$g_\phi = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} P_\phi^t r_\phi = P_\phi^* r_\phi$$

is constant; hence for unichain MDPs, we'll simply let $g_\phi \triangleq g_\phi(x)$, $x \in \mathbb{X}$. Since there exists an optimal stationary policy, this implies that the optimal gain g^* is also constant. Hence the first canonical equation (3) is redundant, and we're left with the *unichain optimality equation*

$$\begin{aligned} g^* + h^*(x) &= r(x, \phi^*(x)) + \sum_{y\in\mathbb{X}} p(y|x, \phi^*(x)) h^*(y) \\ &= \max_{a\in\mathbb{A}(x)} \{r(x,a) + \sum_{y\in\mathbb{X}} p(y|x,a) h^*(y)\}, \quad x \in \mathbb{X}. \end{aligned} \tag{8}$$

Before describing the policy iteration algorithm for unichain average-reward MDPs, we first consider how to evaluate the gain of a given stationary policy. To this end, let $\phi \in \Pi^S$, let e denote the vector of all ones, and suppose the constant c and the vector u satisfy

$$ce + (I - P_\phi)u = r_\phi. \tag{9}$$

Then, since $P_\phi^* P_\phi = P_\phi^*$ and P_ϕ^* is row stochastic (because each row of P_ϕ^* defines the stationary distribution of P_ϕ), multiplying both sides of (9) by P_ϕ^* gives $ce = P_\phi^* r_\phi = g_\phi e$, and hence $c = g_\phi$. Hence to determine the gain g_ϕ of any $\phi \in \Pi^S$, it suffices to find any pair (c, u) satisfying (9). To verify that a solution exists for any $\phi \in \Pi^S$, one can either set $u(x_0) = 0$ for some $x_0 \in \mathbb{X}$ and solve for the remaining variables, or check that $c = g_\phi$ and $u = (Z_\phi - P_\phi^*) r_\phi \triangleq h_\phi$, where $Z_\phi \triangleq (I - P_\phi + P_\phi^*)^{-1}$ is the *fundamental matrix* associated with ϕ, satisfies (9). Incidentally, the vector h_ϕ is called the *bias* of ϕ. In particular, here the expected total reward earned in n steps under ϕ starting from state x can be written as

$$n g_\phi + h_\phi(x) + o(1),$$

where $o(1)$ is a function that goes to zero as $n \to \infty$; hence the bias can be used to differentiate between policies with the same gain, and in fact can be used to demonstrate that the policy iteration algorithm described below terminates after a finite number of iterations.

We now describe the policy iteration algorithm for unichain average-reward MDPs. Recall that the operator U is defined for $u : \mathbb{X} \to \mathbb{R}$ as

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

Also, for $\phi \in \Pi^S$ define U_ϕ for $u : \mathbb{X} \to \mathbb{R}$ by

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

Algorithm 3 Policy Iteration (Unichain MDP, Average Rewards)

1: Select any $\phi \in \Pi^S$.
2: Evaluate ϕ by finding any (g_ϕ, u_ϕ) satisfying $g_\phi e + u_\phi = r_\phi + P_\phi u_\phi$.
3: **while** $Uu_\phi > g_\phi e + u_\phi$ **do**
4: Let ψ be any policy satisfying $U_\psi u_\phi > g_\phi + u_\phi$.
5: Set $\phi = \psi$.

The policy iteration algorithm is given below:

It can be shown (see e.g. Puterman [22, §8.6.2-8.6.3]) that the gain of successive policies generated by Algorithm 3 monotonically increases, and that $g_\psi = g_\phi$ implies that $h_\psi > h_\phi$, where h_ϕ is the bias of the policy ϕ. Since the number of stationary policies is finite when the number of states and actions is finite, this means that the algorithm terminates after a finite number of iterations.

In particular, at termination we have $Uu_\psi \leq g_\psi e + u_\psi$, which means that

$$U_\phi u_\psi = r_\phi + P_\phi u_\psi \leq g_\psi e + u_\psi, \quad \text{for all } \phi \in \Pi^S.$$

Multiplying both sides by P_ϕ^*, we see that this means $g_\phi e = P_\phi^* r_\phi \leq g_\psi e$ for all $\phi \in \Pi^S$. Since a stationary average-reward optimal policy exists when the number of states and actions is finite, this in turn implies that the terminal policy ψ is optimal.

Note that, analogously to policy iteration for discounted-reward MDPs, we may have a choice as to the updated policy ψ to use in line 4 of Algorithm 3. In fact, given a rule for selecting ψ, it turns out that policy iteration for unichain average-reward MDPs is also equivalent to applying the simplex method with a certain pivoting rule to a certain linear program.

1.3.5 Linear Programming: Discounted Rewards

We now turn to the linear programming (LP) formulation of the problem of finding an optimal policy for discounted-reward MDPs. To begin, suppose the vector $v \in \mathbb{R}^n$ satisfies

$$v(x) \geq r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) v(y), \quad \text{for all } x \in \mathbb{X},\ a \in \mathbb{A}(x);$$

such a vector v is called *β-superharmonic*. Then $v \geq Tv$, which implies by induction that $v \geq T^n v$ for all $n \in \mathbb{N}$. The Banach fixed-point theorem in turn implies that $v \geq V_\beta$; since $V_\beta = TV_\beta$, it follows that the value function is the smallest β-superharmonic vector. Hence to find V_β, we can solve the linear program $\min\{\sum_{x \in \mathbb{X}} v(x) : v \text{ is } \beta\text{-superharmonic}\}$. Letting e denote the vector of all ones, letting the $m \times n$ matrices J and P be defined by $[J]_{xa,y} = \delta_{xy}$ (δ_{xy} is defined to be 1 if $x = y$ and 0 otherwise) and $[P]_{xa,y} = p(y|x,a)$, and letting the m-vector r be defined by $[r]_{xa} = r(x,a)$, this LP can be written as

$$\begin{aligned} \text{minimize} \quad & e^T v \\ \text{such that} \quad & (J - \beta P) v \geq r, \end{aligned} \tag{D_β}$$

where the superscript T denotes the transpose, and all vectors are assumed to be column vectors unless stated otherwise. It turns out that the policy iteration algorithm for discounted-reward MDPs can be viewed as applying a simplex method to the dual of (D_β)

$$\begin{array}{ll} \text{maximize} & \rho^T r \\ \text{such that} & \rho^T(J-\beta P) = e^T, \\ & \rho \geq 0. \end{array} \qquad (\mathrm{P}_\beta)$$

In particular, first note that any stationary policy ϕ furnishes us with an initial feasible basis for (P_β); selecting the rows of $J-\beta P$ corresponding to the actions used by ϕ, we obtain the nonsingular matrix $I-\beta P_\beta$, and letting ρ_ϕ denote the vector containing the components of ρ corresponding to the actions selected by ϕ,

$$\rho_\phi = (I-\beta P_\phi^T)^{-1} e = \sum_{t=0}^{\infty} \beta^t (P_\phi^T)^t e \geq \beta^0 (P_\phi^T)^0 e = e > 0. \qquad (10)$$

Next, letting r_ϕ denote the vector containing the elements of r corresponding to the actions selected by ϕ, the corresponding reduced cost vector $\overline{c}_\phi$ is given by

$$\overline{c}_\phi = r - (J-\beta P)(I-\beta P_\phi)^{-1} r_\phi.$$

Since $(I-\beta P_\phi)^{-1} r_\phi = v_\phi$, this means

$$\begin{aligned} \overline{c}_\phi(x,a) &= r(x,a) - \sum_{y\in\mathbb{X}} (\delta_{xy} - \beta p(y|x,a)) v_\phi(y) \\ &= r(x,a) + \beta \sum_{y\in\mathbb{X}} p(y|x,a) v_\phi(y) - v_\phi(x), \quad \text{for } x \in \mathbb{X},\ a \in \mathbb{A}(x). \end{aligned}$$

Hence $\overline{c}_\phi(x,a) > 0$ (i.e. the nonbasic variable $\rho(x,a)$ is eligible to enter the basis) iff. $Tv_\phi(x) \geq r(x,a) + \sum_{y\in\mathbb{X}} p(y|x,a) v_\phi(y) > v_\phi(x)$ (i.e. the policy ϕ can be improved by setting $\psi(x) = a$). Further, $\overline{c}_\phi(x,a) \leq 0$ for all $x \in \mathbb{X}$ and $a \in \mathbb{A}(x)$ (i.e. the basic feasible solution induced by ϕ is optimal) iff. $Tv_\phi = v_\phi$ (i.e. ϕ is optimal). Further, one can show that there is a 1-1 correspondence between the set of basic feasible solutions to (P_β) and the set of stationary policies; see e.g. Puterman [22, §6.9.2]. Recalling that the updated policy ψ in the policy iteration algorithm may update the action in more than one state, it follows that we are justified in performing block pivots when applying the simplex method to (P_β).

1.3.6 Linear Programming: Average Rewards

We now consider the LP formulation of the problem of finding an optimal policy for unichain average-reward MDPs. Analogously to the discounted case, the value g is *superharmonic* if there exists a vector $u \in \mathbb{R}^n$ such that

$$g + u(x) \geq r(x,a) + \sum_{y\in\mathbb{X}} p(y|x,a) u(y), \quad \text{for all } x \in \mathbb{X},\ a \in \mathbb{A}(x).$$

Since the unichain optimality equation (8) has a solution (g^*, h^*), where g^* is the optimal gain, it follows that g^* is the solution to the linear program

$$\begin{aligned} &\text{minimize} \quad g \\ &\text{such that} \quad g + u(x) - \sum_{y \in \mathbb{X}} p(y|x,a)u(y) \geq r(x,a), \quad x \in \mathbb{X},\ a \in \mathbb{A}(x). \end{aligned} \tag{D}$$

The dual of (D) is

$$\begin{aligned} \text{maximize} \quad & \sum_{x \in \mathbb{X}} \sum_{a \in \mathbb{A}(x)} r(x,a)\rho(x,a) \\ \text{such that} \quad & \sum_{x \in \mathbb{X}} \sum_{a \in \mathbb{A}(x)} (\delta_{xy} - p(y|x,a))\rho(x,a) = 0, \quad y \in \mathbb{X}, \\ & \sum_{x \in \mathbb{X}} \sum_{a \in \mathbb{A}(x)} \rho(x,a) = 1, \\ & \rho(x,a) \geq 0, \quad x \in \mathbb{X},\ a \in \mathbb{A}(x). \end{aligned} \tag{P}$$

Here every stationary policy ϕ corresponds to a basic feasible solution ρ_ϕ to (P) by letting $\rho(x, \phi(x))$ be basic for each $x \in \mathbb{X}$ and letting the remaining variables be nonbasic. The vector of basic variable values $b_\phi = [\rho_\phi(x, \phi(x))]_{x \in \mathbb{X}}$ corresponding to such a basic feasible solution is then precisely the stationary distribution of the Markov chain with transition matrix P_ϕ. This is because b_ϕ is a solution to $b^T(I - P_\phi) = 0$ such that $b^T e = 1$, whose unique solution is the stationary distribution of P_ϕ if the MDP is unichain and aperiodic. However, unlike the discounted case, there is no 1-1 correspondence between basic feasible solutions and stationary policies; see e.g. Puterman [22, §8.8.2]. The connection between the simplex method applied to (P) also requires the use of a modified rule for exiting basic variables; see e.g. Denardo [2].

2 Review of Complexity Results

Under both the discounted and average-reward criteria, the problem of finding an optimal policy is tractable. Since both problems can be formulated as linear programs, they can both be solved in polynomial time (see e.g. Khachiyan [14] and Karmarkar [13]). In fact, it was known at least since the late 1980s that under the discounted-reward criterion, both Howard's policy iteration and value iteration also return an optimal policy in polynomial time (Meister & Holzbaur [18] and Tseng [24], respectively). Also of note is that in 1987, Papadimitriou & Tsitsiklis [20] showed that the problem of finding an optimal policy under both the discounted and average-reward criteria is P-complete, by showing that it is P-hard via a reduction from the circuit value problem. This means that there does not exist an efficient parallel algorithm[1] for solving these

[1] i.e. takes time polynomial in the logarithm of the bit-size of the input using a number of processors polynomial in the bit-size of the input.

two problems unless there is one for every problem in P; whether or not the latter is true is unknown.

Hence, interest has since turned towards the possiblity of *strongly* polynomial algorithms, defined in Section 2.1 below, for finding optimal policies under both the discounted and average-reward criteria. In recent years there have been both positive and negative results. On the one hand, for certain problems new strongly polynomial algorithms have been developed (Zadorojniy et al. [28], Ye [26]) and variants of policy iteration were shown to run in strongly polynomial time (Ye [27], Even & Zadorojniy [3], Hansen et al. [9], Post & Ye [21], Feinberg & Huang [5], Scherrer [23], Akian & Gaubert [1]). On the other hand, Feinberg & Huang [6] showed that under the discounted-reward criterion value iteration is not strongly polynomial, and problems were constructed that forced certain variants of policy iteration to take an exponential number of iterations (Melekopoglou & Condon [19], Fearnley [4], Hollanders et al. [10]). These results are described in the following sections.

2.1 Strongly Polynomial Algorithms

An algorithm for finding an optimal policy for an MDP is *polynomial* if the number of arithmetic operations needed to return an optimal policy is bounded by a polynomial in the total number of actions m and the bit-size L of the input data; note that since the set of actions $\mathbb{A}(x)$ available at any given state is assumed to be nonempty, the number of states n is never larger than m. If the number of arithmetic operations needed is bounded by a polynomial only in m, then the algorithm is *strongly polynomial*. A polynomial algorithm that is not strongly polynomial is called *weakly polynomial*.

2.2 Value Iteration

The two results we describe below pertain to the complexity of value iteration under the discounted-reward criterion.

2.2.1 Positive Results

On the positive side, Tseng [24] showed in 1990 that if the discount factor β and transition probabilities $p(y|x,a)$ are rational numbers and the one-step rewards $r(x,a)$ are all integers, then value iteration is guaranteed to return an optimal policy after a weakly polynomial number of iterations for a fixed discount factor.

In particular, let u^k denote the k^{th} vector generated by value iteration (Algorithm 1), where u^0 denotes the initial vector, and let δ be the smallest positive integer such that for all $x, y \in \mathbb{X}$ and $a \in \mathbb{A}(x)$, $\delta\beta$ and $\delta p(y|x,a)$ are integers, $|r(x,a)| \leq \delta$, and $|u^0(x)| \leq \delta$. Then, letting $K \triangleq \max_{x,a} |r(x,a)|$, after

$$k \leq \hat{k} = \left\lceil \frac{\log(2\delta^{2n+2} n^n (\|u^0\| + K/(1-\beta)))}{\log(1/\beta)} \right\rceil \tag{11}$$

iterations, any stationary policy ψ satisfying $T_\psi u^k = Tu^k$ is such that $T_\psi V_\beta = V_\beta$; since v_ψ is the unique fixed point of T_ψ, this means $V_\beta = v_\psi$, i.e. ψ is optimal. Since both $\|u^0\|$ and K are at most δ, and at most a constant times $mn\log(\delta)$ bits are needed to write down the input data, the iteration bound (11) is weakly polynomial for a fixed discount factor. The proof of the bound (11) involves showing that if $\|V_\beta - u^k\| < 1/(2\delta^{2n+2}n^n)$ and $a \in \mathbb{A}(x)$ is not an optimal action, then $\psi(x) \neq a$.

2.2.2 Negative Results

While Tseng showed that for discounted rewards, value iteration returns an optimal policy in weakly polynomial time given rational input data, Feinberg & Huang [6] showed via a simple 3-state example that if the inputs are not assumed to be rational, then the number of arithmetic operations needed to return an optimal policy can grow arbitrarily quickly as a function of the number of actions m. This implies that the algorithm is not strongly polynomial.

Since the example is very simple, we describe it in detail below. Consider an arbitrary increasing sequence $\{M_i\}_{i=1}^\infty$ of natural numbers. Let the state space be $\mathbb{X} = \{1, 2, 3\}$, and for a natural number k let the action space be $\mathbb{A} = \{0, 1, \ldots, \ell\}$. Let $\mathbb{A}(1) = \mathbb{A}$, $\mathbb{A}(2) = \{0\}$ and $\mathbb{A}(3) = \{0\}$ be the sets of actions available at states 1, 2, and 3, respectively. The transition probabilities are given by $p(2|1,i) = p(3|1,0) = p(2|2,0) = p(3|3,0) = 1$ for $i = 1, \ldots, k$. Finally, the one-step rewards are given by $r(1,0) = r(2,0) = 0$, $r(3,0) = 1$, and

$$r(1,i) = \frac{\beta}{1-\beta}(1 - \exp(-M_i)), \quad i = 1, \ldots, \ell.$$

Figure 1 below illustrates such an MDP for $\ell = 2$.

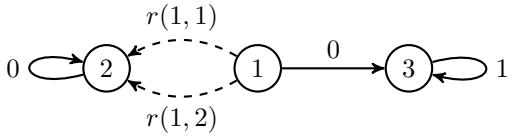

Figure 1: The solid arcs correspond to transitions associated with action 0, and dashed arcs correspond to the remaining actions. The number next to each arc is the reward associated with the corresponding action.

For this MDP each policy is defined by an action selected at state 1. Note that if action $i \in \{1, \ldots, \ell\}$ is selected, then the total discounted reward starting from state 1 is simply $r(1,i)$; if action 0 is selected, the corresponding total discounted reward is $\beta/(1-\beta)$. Since

$$r(1,i) = \frac{\beta}{1-\beta}(1 - \exp(-M_i)) < \frac{\beta}{1-\beta}$$

for each $i = 1, \ldots, \ell$, action 0 is the unique optimal action in state 1.

Now, setting $u^0 \equiv 0$, for $k = 0, 1, \ldots$, we have

$$u^{k+1}(x) = Tu^k(x) = \max_{a \in \mathbb{A}(x)} \{r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) u^k(y)\}, \quad x \in \mathbb{X},$$

and letting ψ^k denote the policy obtained if the algorithm terminates after k iterations, we have that $T_{\phi^k} u^k = Tu^k = u^{k+1}$, i.e.

$$\psi^k(x) \in \arg\max_{a \in \mathbb{A}(x)} \{r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) u^k(y)\}, \quad x \in \mathbb{X}.$$

For this MDP, since the M_i's are increasing in i, we have for $k = 0, 1, \ldots$

$$\begin{aligned}
u^{k+1}(1) &= \max\left\{ \frac{\beta(1-\beta^k)}{1-\beta}, \frac{\beta(1-\exp(-M_\ell))}{1-\beta} \right\} \\
u^{k+1}(2) &= 0, \\
u^{k+1}(3) &= \frac{1-\beta^{k+1}}{1-\beta},
\end{aligned}$$

which means that

$$\psi^k(1) = \begin{cases} \ell, & \text{if } k < M_\ell/(-\ln\beta), \\ 0, & \text{if } k > M_\ell/(-\ln\beta). \end{cases}$$

Hence more than $M_k/(-\ln\beta)$ iterations are needed to select the optimal action 0 in state 1. For example, if $M_k = 2^k$, then since there are a total of $m = k+3$ actions, at least $C \cdot 2^m/(-\ln\beta)$ iterations are needed, where $C = 2^{-3}$.

2.3 Policy Iteration

The majority of the complexity results considered here pertain to the policy iteration algorithm, to which we now turn.

2.3.1 Positive Results

Meister & Holzbaur [18] showed in 1986 that given rational input data, Howard's policy iteration algorithm is weakly polynomial. In particular, recalling that L denotes the bit-size of the input data, the number of iterations needed to return an optimal policy is at most a constant times $n \log L / \log(1/\beta)$. Since each iteration involves solving the linear system $(I - \beta P_\phi)u = r_\phi$ for some $\phi \in \Pi^S$ and finding a policy ψ such that $T_\psi v_\phi = Tv_\phi$, which can be done with $O(n^3 + n^2 m)$ arithmetic operations, the algorithm is weakly polynomial. As was noted in Section 1.3.3, their proof depends on the observation that the sequence of policies $\{\phi^k\}_{k \geq 0}$ generated by the algorithm is such that

$$\|V_\beta - v_{\phi^{k+1}}\| \leq \beta \|V_\beta - v_{\phi^k}\|,$$

which, since $v_{\phi^k} \leq v_{\phi^{k+1}} \leq V_\beta$ and recalling that $K = \max_{x,a} |r(x,a)|$, implies that

$$\|v_{\phi^{k+1}} - v_{\phi^k}\| \leq \beta^k \frac{2K}{1-\beta}. \tag{12}$$

Next, they show that if $v_{\phi^{k+1}} \neq v_{\phi^k}$, then

$$\|v_{\phi^{k+1}} - v_{\phi^k}\| \geq \frac{1}{N^{2n-1}(1-\beta)^2(1+\beta)^{2n-2}}, \tag{13}$$

where N is the greatest common denominator of the $\beta p(y|x,a)$'s. The result follows by using (12) and (13) to bound the number of iterations until we have $v_{\phi^{k+1}} = v_{\phi^k}$, i.e. until the algorithm terminates with an optimal policy.

For the next two decades, a polynomial iteration bound for policy iteration independent of the bit-size L of the input remained elusive. In 1999, Mansour & Singh [17] showed that Howard's policy iteration takes $O(\overline{m}^n/n)$ iterations to obtain an optimal policy on a problem with at most $\overline{m}$ actions per state; this is still the best-known bound that is independent of both the discount factor β and L, but is only modestly better than the trivial bound of $O(\overline{m}^n)$.

A breakthrough finally came in 2011, when Ye [27] used the linear programming formulation of the problem to show that both Howard's policy iteration and the simplex method using Dantzig's classic pivoting rule need at most

$$(m-n)\left(1 + \frac{n}{1-\beta}\log\left(\frac{n^2}{1-\beta}\right)\right) \tag{14}$$

iterations to return an optimal policy for discounted-reward MDPs. Since each iteration of policy iteration/the simplex method can be done in $O(n^3 + mn)$ arithmetic operations (actually, using the revised simplex method, basis updates can be done using $O(n^2)$ arithmetic operations), this showed for the first time that Howard's policy iteration algorithm and the simplex method with Dantzig's pivoting rule solve discounted-reward MDPs with a fixed discount factor in *strongly polynomial time*. It is interesting to note that in 1972, Klee & Minty [16] showed that the simplex method with Dantzig's rule can require an exponential number of iterations to solve an LP in general.

We now describe the approach Ye used to show that the iteration bound (14) holds. The proof was carried out by considering the behavior of the simplex method on the LP

$$\begin{aligned} \text{maximize} \quad & \rho^T r \\ \text{such that} \quad & \rho^T(J - \beta P) = e^T, \\ & \rho \geq 0. \end{aligned} \tag{P_β}$$

First, he notes that every feasible ρ satisfies

$$e^T \rho = \frac{n}{1-\beta}, \tag{15}$$

and that the values of the basic variables ρ_ϕ of any basic feasible solution satisfy

$$\rho_\phi = (I - \beta P_\phi^T)^{-1} e = \sum_{t=0}^{\infty} \beta^t (P_\phi^T)^t e \geq \beta^0 (P_\phi^T)^0 e = e > 0.$$

Letting

$$\Delta_\phi \triangleq \max_{x,a}\{\overline{c}_\phi(x,a) = r(x,a) + \beta \sum_{y\in\mathbb{X}} p(y|x,a)v_\phi(y) - v_\phi(x)\},$$

the fact that $\rho_\phi \geq e$ implies that under both Howard's policy iteration and the simplex method with Dantzig's pivoting rule, the updated policy ψ improves on the objective value $\rho_\phi^T r_\phi$ of the current policy ϕ by at least Δ_ϕ. This and (15) are used to show that if ϕ^0 is the initial policy and ϕ^k is the k^{th} policy generated by Howard's policy iteration or the simplex method with Dantzig's pivoting rule, and z^* is the optimal objective function value of the LP (P_β), then

$$\frac{z^* - \rho_{\phi^k}^T r_{\phi^k}}{z^* - \rho_{\phi^0}^T r_{\phi^0}} \leq \left(1 - \frac{1-\beta}{n}\right)^k. \tag{16}$$

Then, using strong duality and a well-known strict complementarity result for LPs, he shows that if ϕ^0 is nonoptimal, then for some $x_0 \in \mathbb{X}$ it is never optimal to use the action $\phi^0(x_0)$ and that, if ϕ^k still uses the action $\phi^0(x_0)$ in state x_0, then

$$1 \leq \rho_{\phi^k}(x_0, \phi^k(x_0)) \leq \frac{n^2}{1-\beta}\frac{z^* - \rho_{\phi^k}^T r_{\phi^k}}{z^* - \rho_{\phi^0}^T r_{\phi^0}}, \tag{17}$$

where we recall that the inequality on the left holds for any basic variable for the LP (P_β). The bound (14) follows by using (16) and (17) to show that if ϕ^0 is nonoptimal then some nonoptimal action is permanently eliminated from consideration after at most

$$\frac{n}{1-\beta}\log\left(\frac{n^2}{1-\beta}\right) \tag{18}$$

iterations, and noting that because an optimal policy exists there are at most $m-n$ non-optimal actions that need to be eliminated.

In early 2013, Hansen Miltersen & Zwick [9] improved the iteration bound (14) for Howard's policy iteration to

$$(m-n)\left(1 + \frac{1}{1-\beta}\log\left(\frac{n}{1-\beta}\right)\right), \tag{19}$$

and also showed that the same bound applies to the related *strategy iteration* algorithm for solving discounted two-player turn-based zero-sum games. They do this by improving Ye's estimate (18) of the number of iterations needed in order for a nonoptimal action to be permanently eliminated to

$$\frac{1}{1-\beta}\log\left(\frac{n}{1-\beta}\right)$$

using the property that, if ϕ^0 is the initial policy, then the k^{th} policy generated by Howard's policy iteration algorithm satisfies

$$\|V_\beta - v_{\phi^k}\| \leq \beta^k \|V_\beta - v_{\phi^k}\|.$$

Around the same time, Post & Ye [21] showed that for deterministic MDPs, the number of iterations required by the simplex method with Dantzig's pivoting rule to find an optimal policy is strongly polynomial irrespective of the discount factor. In particular, the time required is proportional to $n^3m^2\log^2 n$. They also showed that if each action involves a distinct discount factor, then the required number of iterations is proportional to $n^5m^3\log^2 n$. Noting that a stationary policy for a deterministic MDP consists of a set of paths and cycles over the states, their analysis involves showing that after a polynomial number of iterations, either a new cycle is created or the algorithm terminates, and then showing that whenever a new cycle is created significant progress towards finding the optimal policy is made.

Ye's result that Howard's policy iteration and the simplex method with Dantzig's pivoting rule is strongly polynomial for a fixed discount factor has also been used to show that certain average-reward problems. In particular, Feinberg & Huang [5] showed in 2013 that if there is a state to which the process transitions with probability at least $\gamma > 0$ under any action, then an average-reward optimal policy can be found in strongly polynomial time for a fixed γ. This is because such a problem can be reduced to a discounted-reward problem with discount factor $\beta = 1 - \gamma$ by setting the transition probabilities to $\tilde{p}$, where

$$\tilde{p}(j|i,a) = \begin{cases} p(j|i,a)/(1-\gamma), & \text{if } j \neq i^*, \\ (p(i^*|i,a)-\gamma)/(1-\gamma), & \text{if } j = i^*, \end{cases}$$

and keeping the state space, action spaces, and rewards the same. Noting that the original MDP is unichain, it is easily verified that applying the unichain policy iteration algorithm for average-reward MDPs to the original MDP, where u_ϕ is determined in each step by setting $u_\phi(\ell) = 0$ for some $\ell \in \mathbb{X}$, is in fact equivalent to applying policy iteration to the associated discounted-reward problem.

Another improvement to the iteration bound for Howard's policy iteration algorithm for discounted-reward MDPs, as well as for the simplex method with Dantzig's pivoting rule, came in the middle of 2013 when Scherrer [23] showed that Howard's policy iteration needs at most

$$(m-n)\left(1 + \frac{1}{1-\beta}\log\left(\frac{1}{1-\beta}\right)\right)$$

iterations, while the simplex method with Dantzig's pivoting rule needs at most

$$n(m-n)\left(1 + \frac{2}{1-\beta}\log\left(\frac{1}{1-\beta}\right)\right)$$

iterations. We describe his proof of the bound for Howard's policy iteration below; his proof of the bound for the simplex method is more involved. First, he shows that if ϕ^0 is the initial policy, then the k^{th} policy ϕ^k satisfies

$$\|V_\beta - T_{\phi^k}V_\beta\| \leq \frac{\beta^k}{1-\beta}\|V_\beta - T_{\phi^0}V_\beta\|. \tag{20}$$

Then, he notes that if ϕ^0 is not optimal, then $V_\beta \neq T_{\phi^0} V_\beta$, and so there exists a state x_0 such that

$$V_\beta(x_0) - T_{\phi^0} V_\beta(x_0) = \|V_\beta - T_{\phi^0} V_\beta\| > 0.$$

In particular, the action selected by ϕ^0 in state x_0 is not optimal, because it is not conserving. From (20), we immediately have that the k^{th} policy generated after ϕ^0 is such that

$$V_\beta(x_0) - T_{\phi^k} V_\beta(x_0) \leq \frac{\beta^k}{1-\beta}(V_\beta(x_0) - T_{\phi^k} V_\beta(x_0)),$$

which means that if $k > \log(1/(1-\beta))/(1-\beta)$, then $\phi^k(x_0) \neq \phi^0(x_0)$, i.e. the nonoptimal action selected by ϕ^0 in state x_0 will not be used again after $\log(1/(1-\beta))/(1-\beta)$ iterations.

Most recently, Akian & Gaubert [1] used methods from nonlinear Perron-Frobenius theory to show that, if the MDP has a state that is recurrent under every stationary policy, then Howard's policy iteration algorithm terminates in strongly polynomial time. Their proof involved applying a transformation that does not affect the sequnce of policies generated by the algorithm.

2.3.2 Negative Results

On the negative side, Melekopoglou & Condon [19] showed in 1994 that four pivoting rules for the simplex method, in particular the least-index rule and the best improvement rule, can require a number of iterations exponential in the number of states to obtain the optimal policy under both the discounted and average-reward criteria.

In addition, in 2010 Fearnley [4] exhibited a unichain MDP on which the policy iteration algorithm for average rewards takes an exponential number of iterations. The example is quite elaborate, and involves associating each stationary policy with the state of a binary counter and showing that policy iteration must consider each state of the binary counter before arriving at the optimal policy.

Most recently, Hollanders Delvenne & Jungers [10] showed in 2012 how Fearnley's example could be adapted to show, via a suitable perturbation, that Howard's policy iteration algorithm can take an exponential number of iterations if the discount factor is part of the input.

2.4 Other Algorithms

We note that, in addition to the new complexity results for classical algorithms, two new algorithms have recently been developed. One was an interior-point method proposed by Ye [26] in 2005 for discounted-reward MDPs, and was the first strongly polynomial time algorithm for this problem when the discount factor is fixed. Its iteration bound, however, was worse than the bound for policy iteration obtained by Ye [27] in 2011.

Another new algorithm was developed by Zadorojniy Even & Schwartz [28] in 2009 for solving controlled random walks under both the discounted and average-reward criteria. The total running time under both criteria was shown to be at most a constant times $n^4\overline{m}^2$, where $\overline{m}$ is the maximum number of actions per state. In 2012, Even & Zadorojniy [3] showed that this algorithm is in fact the simplex method with the Gass-Saaty shadow vertex pivoting rule, and using this representation were able to improve the running time by a factor of n.

3 Future Directions

We now describe some possible directions for future research.

3.1 The Majorant Condition

In this section we consider the average-reward criterion. A condition on the transition function that is related to the condition considered in Feinberg & Huang [5] is where there exists a number $q(y)$ for each state y such that

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

We'll say that an MDP satisfying this condition has a *majorant*. Once fact about such an MDP is that it is unichain; this is because if some stationary policy has more than one recurrent class, then the sum above has to be at least 2 if the condition on the left is satisfied.

Another fact that makes such an MDP attractive is that it shares some similarities with the discounted-reward problem. For instance, to find an optimal policy it suffices to find the fixed point of a contraction mapping. In particular, let $\beta \triangleq \sum_{y \in \mathbb{X}} q(y) - 1$, let $\tilde{p}(y|x,a) = \beta^{-1}(q(y) - p(y|x,a))$ for each $x, y \in \mathbb{X}$ and $a \in \mathbb{A}(x)$, and let the operator $\mathcal{U}$ be defined for $u : \mathbb{X} \to \mathbb{R}$ as

$$\mathcal{U}u(x) = \max_{a \in \mathbb{A}(x)} \{r(x,a) - \beta \sum_{y \in \mathbb{X}} \tilde{p}(y|x,a)u(y)\}, \quad x \in \mathbb{X}.$$

Using the same technique used in the discounted case, one can show that $\mathcal{U}$ is a contraction mapping with modulus β, which implies that it has a unique fixed point u^*, i.e.

$$u^*(x) = \max_{a \in \mathbb{A}(x)} \{r(x,a) - \beta \sum_{y \in \mathbb{X}} p(y|x,a)u^*(y)\}, \quad x \in \mathbb{X}.$$

Using the definition of β, this can be rewritten as

$$\sum_{y \in \mathbb{X}} q(y)u^*(y) + u^*(x) = \max_{a \in \mathbb{A}(x)} \{r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a)u^*(y)\}, \quad x \in \mathbb{X}, \quad (21)$$

which shows that $(\sum_{y\in\mathbb{X}} q(y)u^*(y), u^*)$ satisfies the unichain optimality equations. Hence if the MDP has a majorant, to find an optimal policy it suffices to find the unique u^* such that $u^* = \mathcal{U}u^*$; the optimal gain is then $g^* = \sum_{y\in\mathbb{X}} q(y)u^*(y)$, and any policy attaining the maximum on the right-hand side of (21) is optimal.

Of course, in this case value iteration is applicable, and has the same convergence rate as in the discounted case. Also, one can show that given any stationary policy ϕ, the system

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

has a unique solution u_ϕ, and that the gain under ϕ is $g_\phi = \sum_{y\in\mathbb{X}} q(y)u_\phi(y)$. From this we obtain a version of policy iteration that follows the general version for unichain MDPs given above, except we perform a "value determination" step analogously to the discounted case to get u_ϕ. In fact, using β and $\tilde{p}$ defined above, this corresponds to running policy iteration for discounted-reward MDPs using a negative discount factor. A possibility would then be to try and adapt Scherrer's [23] proof technique to obtain a bound on the running time of policy iteration for such a problem.

One can also write down a linear program for this problem, which resembles the primal LP in the discounted case:

$$\begin{aligned}
\text{maximize} \quad & \sum_{x\in\mathbb{X}}\sum_{a\in\mathbb{A}(x)} r(x,a)\rho(x,a) \\
\text{such that} \quad & \sum_{a\in\mathbb{A}(y)} \rho(y,a) + \sum_{x\in\mathbb{X}}\sum_{a\in\mathbb{A}(x)} (q(y) - p(y|x,a))\rho(x,a) = q(y), \quad y\in\mathbb{X}, \\
& \rho(x,a) \geq 0, \quad x\in\mathbb{X},\ a\in\mathbb{A}(x).
\end{aligned}$$

One possiblity that immediately suggests itself is to try and modify Ye's approach to apply to this LP, in particular to find a postive lower bound on the values any positive basic variable; following Kitahara & Mizuno's [15] generalization of Ye's [27] technique, and demonstrating that Dantzig's pivoting rule does not cycle on this LP by linking it with unichain policy iteration, this would provide a bound analogous to Ye's on the number of iterations needed. Another approach is to somehow split the variables in such a way that the value of any positive basic variable is bounded below by a positive number.

One aspect of this problem that may suggest that policy iteration/linear programming may perform poorly is that there may not be a state that is recurrent under all stationary policies; in particular, this means that the result of Akian & Gaubert [1] is not applicable. For example, in the following MDP has a majorant, but every state is transient under some stationary policy: let $\mathbb{X} = \{1,2,3\}$, $\mathbb{A} = \{1,2\} = \mathbb{A}(1) = \mathbb{A}(2) = \mathbb{A}(3)$, and let

$$
\begin{array}{lll}
p(1|1,1) = 1/2, & p(2|1,1) = 1/2, & p(3|1,1) = 0; \\
p(1|1,2) = 1/2, & p(2|1,2) = 0, & p(3|1,2) = 1/2; \\
p(1|2,1) = 0, & p(2|2,1) = 1/2, & p(3|2,1) = 1/2; \\
p(1|2,2) = 1/2, & p(2|2,2) = 1/2, & p(3|2,2) = 0; \\
p(1|3,1) = 0, & p(2|3,1) = 1/2, & p(3|3,1) = 1/2; \\
p(1|3,2) = 1/2, & p(2|3,2) = 0, & p(3|3,2) = 1/2.
\end{array}
$$

A majorant for this MDP is $q(1) = q(2) = q(3) = 1/2$, but state 1 is transient under the policy $\phi(1) = \phi(2) = \phi(3) = 1$ and states 2 and 3 are transient under the policy $\phi(1) = 1$ & $\phi(2) = \phi(3) = 2$. Of course, an MDP with a majorant on which policy iteration takes exponential time would be of interest; we note that Fearnley's [4] example does not have a majorant.

3.2 Communicating MDPs

Another possibility is to investigate the implications of the condition that the MDP is *communicating*, i.e. where for every pair x, y of states there is a stationary policy ϕ such that x is accessible from y under ϕ in $n \geq 1$ steps. One suggestion that the communicating condition may be desirable from a complexity standpoint is that checking whether an MDP is communicating can be done in polynomial time, as was shown by Kallenberg [12], while Tsitsiklis [25] showed that checking whether an MDP is unichain is NP-hard.

3.3 Complexity of Simplex Pivoting Rules

We have already seen that the choice of a pivoting rule for the simplex method can have important consequences for the performance of the algorithm on certain kinds of MDPs, e.g. the Gass-Saaty rule makes the simplex method strongly polynomial for controlled random walks [3] and Dantzig's rule makees it strongly polynomial when the discount factor is fixed [27], while the least index and best improvement rules can be exponential [19]. In addition, the LP formulation of an MDPs was recently used by Friedmann [7] to obtain a subexponential lower bound for Zadeh's pivoting rule, and by Friedmann Hansen & Zwick [8] to obtain subexponential lower bounds for two randomized pivoting rules.

References

[1] M. Akian and S. Gaubert. Policy iteration for perfect information stochastic mean payoff games with bounded first return times is strongly polynomial. *Preprint*, 2013. `http://arxiv.org/abs/1310.4953v1`.

[2] E. Denardo. A markov decision problem. In T. C. Hu and S. M. Robinson, editors, *Mathematical Programming*, pages 33–68. Academic Press, New York, 1973.

[3] G. Even and A. Zadorojniy. Strong polynomiality of the gass-saaty shadow-vertex pivoting rule for controlled random walks. *Annals of Operations Research*, 201:159–167, 2012.

[4] J. Fearnley. Exponential lower bounds for policy iteration. In S. Abramsky et al., editor, *Automata, Languages and Programming; 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 551–562. Springer, Berlin, 2010.

[5] E. A. Feinberg and J. Huang. Strong polynomiality of policy iterations for average-cost mdps modeling replacement and maintenance problems. *Operations Research Letters*, 41:249–251, 2013.

[6] E. A. Feinberg and J. Huang. The value iteration algorithm is not strongly polynomial for discounted dynamic programming. *Operations Research Letters*, 2014. `http://dx.doi.org/10.1016/j.orl.2013.12.011`.

[7] O. Friedmann. A subexponential lower bound for zadehs pivoting rule for solving linear programs and games. In O. Gunluk and G. J. Woeginger, editors, *Integer Programming and Combinatoral Optimization*, pages 192–206. Springer, 2011.

[8] O. Friedmann, T. D. Hansen, and U. Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. *Proceedings of the 43rd annual ACM symposium on Theory of computing*, 2011.

[9] T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM*, 60(1):Article 1, 16 pages, 2013.

[10] R. Hollanders, J. Delvenne, and R. M. Jungers. The complexity of policy iteration is exponential for discounted markov decision processes. *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on.*, 2012.

[11] R. A. Howard. *Dynamic Programming and Markov Processes*. The MIT Press, Cambridge, MA, 1960.

[12] L. C. M. Kallenberg. Classification problems in mdps. In Z. Hou et al., editor, *Markov Processes and Controlled Markov Chains*, pages 151–165. Kluwer Academic Publishers, 2002.

[13] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[14] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244:1086–1093, 1979.

[15] T. Kitahara and S. Mizuno. A bound for the number of different basic solutions generated by the simplex method. *Mathematical Programming*, 137:579–586, 2011.

[16] V. Klee and G. J. Minty. How good is the simplex method? In O. Shisha, editor, *Inequalities-III*, pages 159–175. Academic Press, New York, 1972.

[17] Y. Mansour and S. Singh. On the complexity of policy iteration. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 401–408, 1999.

[18] U. Meister and U. Holzbaur. A polynomial time bound for howard's policy improvement algorithm. *OR Spektrum*, 8:37–40, 1986.

[19] M. Melekopoglou and A. Condon. On the complexity of the policy improvement algorithm for markov decision processes. *ORSA Journal on Computing*, 6(2):188–192, 1994.

[20] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[21] I. Post and Y. Ye. The simplex method is strongly polynomial for deterministic markov decision processes. *to appear in Mathematics of Operations Research*, 2014.

[22] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, 1994.

[23] B. Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 386–394. NIPS Foundation, Inc., 2013.

[24] P. Tseng. Solving h-horizon, stationary markov decision problems in time proportional to log(h). *Operations Research Letters*, 9:287–297, 1990.

[25] J. N. Tsitsiklis. Np-hardness of checking the unichain condition in average cost mdps. *Operations Research Letters*, 35:319–323, 2007.

[26] Y. Ye. A new complexity result on solving the markov decision problem. *Mathematics of Operations Research*, 30(3):733–749, 2005.

[27] Y. Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

[28] A. Zadorojniy, G. Even, and A. Shwartz. A strongly polynomial algorithm for controlled queues. *Mathematics of Operations Research*, 34(4):992–1007, 2009.