

On Policy Iteration as a Newton’s Method and Polynomial Policy Iteration Algorithms

Omid Madani

Department of Computing Science
University of Alberta
Edmonton, AL
Canada T6G 2E8
madani@cs.ualberta.ca

Abstract

Policy iteration is a popular technique for solving Markov decision processes (MDPs). It is easy to describe and implement, and has excellent performance in practice. But not much is known about its complexity. The best upper bound remains exponential, and the best lower bound is a trivial $\Omega(n)$ on the number of iterations, where n is the number of states.

This paper improves the upper bounds to a polynomial for policy iteration on MDP problems with special graph structure. Our analysis is based on the connection between policy iteration and Newton’s method for finding the zero of a convex function. The analysis offers an explanation as to why policy iteration is fast. It also leads to polynomial bounds on several variants of policy iteration for MDPs for which the linear programming formulation requires at most two variables per inequality (MDP(2)). The MDP(2) class includes deterministic MDPs under discounted and average reward criteria. The bounds on the run times include $O(mn^2 \log m \log W)$ on MDP(2) and $O(mn^2 \log m)$ for deterministic MDPs, where m denotes the number of actions and W denotes the magnitude of the largest number in the problem description.

1 Introduction

Markov decision processes offer a clean and rich framework for problems of control and decision making under uncertainty [BDH99; RN95]. A set of central problems in this family is the fully observable Markov decision problems under infinite-horizon criteria [Ber95]. We refer to these as MDP problems in this paper. Not only are the MDP problems significant on their own, but solutions to these problems are used repeatedly in solving problem variants such as stochastic games and partially observable MDPs [Con93; HZ01]. In an MDP model, the system is in one of a finite set of states at any time point. In each state an agent has a number of actions to choose from. Execution of an action gives the agent a reward and causes a stochastic change in the system state. The problem is, given a full description of the system and actions, to find a policy, that is a mapping from states to actions, so that the expected (discounted) to-

tal reward over an indefinite (or infinite) number of action executions is maximized.

Policy improvement is a key technique in solving MDPs. It is simple to describe and easy to implement, and quickly converges to optimal solutions in practice. The improvement method begins with an arbitrary policy, and improves the policy iteratively until an optimal policy is found. In each improvement step, the algorithm changes choice of action for a subset of the states, which leads to an improved policy. These algorithms differ on how the states are picked. In addition to policy improvement algorithms, other methods for solving MDPs include algorithms for linear programming, but variants of policy improvement are preferred due to speed and ease of implementation [Lit96; Han98; GKP01]. We remark that policy improvement can be viewed as a special linear programming solution method [Lit96; Ber95].

Unfortunately, the worst-case bounds on several implementations of policy improvement are exponential [MC94]. The exponential lower bounds have been shown for those policy improvement algorithms that in each iteration attempt to pick a single most promising state to improve, and are established on plausible heuristics for such a choice, such as looking ahead one or a constant number of steps. Let us call these ‘selective’ policy improvement. In a sense, the bounds imply that attempting to be smart about choosing which state to improve can lead to an exponentially long path to the optimal. On the other hand, the policy improvement technique is naturally implemented in a manner in which all states are examined, and any improvable state changes action. We will refer to this variation as policy iteration (PI) (see Section 2.1). It is known that PI is no worse than pseudo-polynomial¹ time [Ber95], while the exponential lower bounds on selective algorithms apply irrespective of the number representation [Lit96]. This suggests that the constraints on how PI advances may be inherently different than those for the selective policy improvement algorithms, and leaves hope for PI. But quantifying the advancement of PI has been difficult. The best upper bound on PI, besides the pseudo-polynomial bound, is also exponential $O(2^n/n)$ [MS99], where n is the number of states and each state has two actions. This upper bound is derived using certain par-

Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

¹That is, polynomial if the numbers are written in unary.

